DOCUMENT RESUME

ED 165 781 IR 006 840

AUTHOR	Branstad, Dennis K., Ed.
TITLE	Computer Security and the Data Encryption Standard. Proceedings of the Conference on Computer Security and the Data Encryption Standard.
INSTITUTION	National Bureau of Standards (DOC), Washington, D.C. Inst. for Computer Sciences and Technology.
REPORT NO	NBS-SP-500-27
PUB DATE	Feb 78
NOTE	131p.; Computer Science & Technology Series; (Conference held at Gaithersburg, Maryland, February 15, 1977)
AVAILABLE FROM	Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402 (Stock No. 003-003-01891-1, $3.00)
EDRS PRICE	MF-$0.83 HC-$7.35 Plus Postage.
DESCRIPTORS	Algorithms; *Codification; *Computers; Conference Reports; *Data Processing; Equipment; Flow Charts; Information Networks; *Security Personnel; *Standards
IDENTIFIERS	*Computer Software; Data Encryption

ABSTRACT

The 15 papers and summaries of presentations in this collection provide technical information and guidance offered by representatives from federal agencies and private industry. Topics discussed include physical security, risk assessment, software security, computer network security, and applications and implementation of the Data Encryption Standard. A list of questions submitted in writing at the conference together with responses prepared by either the speaker, the session chairman, or the editor, are appended. (CMV)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED165781

COMPUTER SCIENCE & TECHNOLOGY:

Computer Security and the Data Encryption Standard

Proceedings of the Conference on Computer Security and the Data Encryption Standard Held at the National Bureau of Standards in Gaithersburg, Maryland on February 15, 1977

Dennis K. Branstad, Editor

Systems and Software Division
Institute for Computer Sciences and Technology
National Bureau of Standards
Washington, D.C. 20234

Sponsored by the
National Bureau of Standards
and the
U.S. Civil Service Commission

BEST COPY AVAILABLE

U.S. DEPARTMENT OF HEALTH, EDUCATION & WELFARE
NATIONAL INSTITUTE OF EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

U.S. DEPARTMENT OF COMMERCE, Juanita M. Kreps, Secretary

Dr. Sidney Harman, Under Secretary

Jordan J. Baruch, Assistant Secretary for Science and Technology

NATIONAL BUREAU OF STANDARDS, Ernest Ambler, Acting Director

Issued February 1978

IR006840

Reports on Computer Science and Technology

The National Bureau of Standards has a special responsibility within the Federal Government for computer science and technology activities. The programs of the NBS Institute for Computer Sciences and Technology are designed to provide ADP standards, guidelines, and technical advisory services to improve the effectiveness of computer utilization in the Federal sector, and to perform appropriate research and development efforts as foundation for such activities and programs. This publication series will report these NBS efforts to the Federal computer community as well as to interested specialists in the academic and private sectors. Those wishing to receive notices of publications in this series should complete and return the form at the end of this publication.

National Bureau of Standards Special Publication 500-27
Nat. Bur. Stand. (U.S.) Spec. Publ. 500-27, 135 pages (Feb. 1978)
CODEN: XNBSAV

Library of Congress Cataloging in Publication Data

Conference on Computer Security and the Data Encryption Standard, Gaithersburg, Md., 1977.

Computer security and the data encryption standard.

(Computer science & technology) (NBS special publication ; 500-27)

1. Computers--Access control--Passwords. I. United States. National Bureau of Standards. II. United States. Civil Service Commission. III. Title. IV. Series. V. Series: United States. National Bureau of Standards. Special publication ; 500-27

QC100.U57 no. 500-27 [QA76.9.A25] 602'.1s [658.4'7] 78-1403

U.S. GOVERNMENT PRINTING OFFICE
WASHINGTON: 1978

For sale by the Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402
Stock No. 003-003-01891-1 Price $3
(Add 25 percent additional for other than U.S. mailing).

FOREWORD

The need for standards in computer security has risen along with the need for improved record keeping practices throughout the Federal Government. The increasing use of computers by Government and private industry for the processing, storing and communication of sensitive as well as valuable data has focused this need and has resulted in an extensive standards development program in this area. The development of the Data Encryption Standard is the first major result of this program.

The Conference on Computer Security and the Data Encryption Standard was organized to disseminate information about the computer security technology available, undergoing development or identified as needing additional effort. The Conference was jointly sponsored by the National Bureau of Standards and the Civil Service Commission to provide this information to Federal supervisors, managers, computer specialists, communication specialists, standards personnel and others with an interest in computer security. The main objective of the Conference was to present a perspective of the use of data encryption as a computer security measure.

The proceedings of this Conference, held at NBS on February 15, 1977, are intended for use by all Federal and private organizations seeking to improve the security of their computer systems. The views expressed in the papers do not necessarily reflect those of the National Bureau of Standards or the Civil Service Commission. The material presents the present state-of-the-art technology for encrypting data and introduces many applications of encryption either implemented or contemplated.

M. Zane Thornton
Acting Director
Institute for Computer
Sciences and Technology

PREFACE

The National Bureau of Standards (NBS) initiated a project in 1972 to develop standards in Computer Systems Security. Among the needs for physical, administrative and technical security measures and procedures, the need for a method of protecting computer data during transmission and storage was identified. As a result of this initial evaluation of security requirements, a project was initiated that led to the publication of the Data Encryption Standard.

The Data Encryption Standard was published on January 15, 1977 and became effective on July 15, 1977. The Standard specifies an algorithm for use by Federal Departments and Agencies in the cryptographic protection of unclassified computer data during transmission or in storage. The algorithm, developed by the International Business Machines Corporation, was one of several algorithms submitted to NBS for consideration as a standard. The algorithm and the proposed standard containing the algorithm were reviewed and coordinated within the standards-making process. The publication of the Standard culminated this process.

The Conference on Computer Security and the Data Encryption Standard was structured to answer many of the questions typically raised concerning computer security and data encryption. Nearly 500 people participated in the Conference. Experts from industry and government were invited to present their views and recommendations on when, where and how to apply encryption to a computer system or network. The presentations were organized to give full coverage to the applications as well as the implementations of the Data Encryption Standard.

These proceedings are structured along the same lines as the Conference. Pictures of the speakers as well as their submitted papers or an edited version of their presentations are included. The Appendix contains the questions that were raised at the conclusion of the Conference as well as the answers prepared by the speakers.

The National Bureau of Standards wishes to thank all who participated in the Conference and helped to make it a success.

S. Jeffery
Conference Chairman

Dennis K. Branstad
Program Chairman

ABSTRACT

These proceedings include papers or summaries of presentations of the fifteen speakers who participated in the Conference on Computer Security and the Data Encryption Standard held at the National Bureau of Standards on February 15, 1977. Representatives from Federal agencies and private industry presented technical information and guidance with respect to computer security and the Data Encryption Standard. Subjects of the papers and presentations include physical security, risk assessment, software security, computer network security, applications and implementation of the Data Encryption Standard. The questions raised at the conference and their answers are included in the proceedings.

Key words: Computer security, cryptography; Data Encryption Standard; encryption; key management; network security.

ACKNOWLEDGEMENTS

The following individuals made significant contributions to the success of the conference and to the publication of these proceedings:

Mrs. Anne Shreve, NBS (Conference Manager)

Mrs. Sara Torrence, NBS (Arrangements Chairman)

Mr. Fred Rao, CSC (Nominations Chairman)

Mrs. Grace Burns, NBS (Institute Liaison)

Mrs. Mary Ellen Crane, NBS (Proceedings Typist)

EDITOR'S COMMENT

All but five of the published papers were received directly from the author(s). The other five papers (Courtney, Rallapalli, Crumb, McDonnell and Tuchman) were edited from the taped presentations made at the Conference. None of the slides shown during these presentations were available for publication in these proceedings.

Certain commercial products are identified in these proceedings in order to specify adequately experimental procedures. In no case does such identification imply recommendation or endorsement by the National Bureau of Standards, nor does it imply that the products or equipment identified are necessarily the best available for the purpose.

CONFERENCE ON COMPUTER SECURITY

AND THE

DATA ENCRYPTION STANDARD

PROGRAM AND INDEX

Page

1. WELCOME AND INTRODUCTION 2

   Mr. S. Jeffery, National Bureau of Standards

2. The Data Encryption Standard in Perspective 4

   Dr. Ruth M. Davis, National Bureau of Standards

3. Major Computer Security Aspects Related to the Data Encryption Standard (Picture of Participants) 14

   3.1 Computer Security Risk Assessment 15

   Mr. Robert H. Courtney, IBM Corporation

   3.2 Data Encryption and its Relationship to Physical Security Planning 13

   Mr. Robert V. Jacobson, Chemical Bank

   3.3 Computer Systems Security and the NBS-DES (Beyond Line Encryption) 25

   Mr. Clark Weissman, System Development Corporation

4. Considerations in Procurement and Use of Data Encryption Devices (Picture of Participants) 37

   4.1 Considerations in Applying an Encryption Device to a Communications Network 38

   Mr. Barrie Morgan, Datotek, Inc.

   4.2 The Management of Encryption Keys 46

   Mr. David J. Sykes, Honeywell Information Systems, Inc.

   4.3 Design and Specification of Cryptographic Capabilities 54

   Mr. Carl M. Campbell, Jr. (Consultant) Interbank Card Association

Page

4.4 A Bit-Slice, 4-Chip Implementation of the Data Encryption Standard 67

Kris Rallapalli, Fairchild Semi-Conductor

5. Applications of the Data Encryption Standard (Picture of Participants) 69

5.1 Federal Reserve Communications Security Project 70

Mr. Howard Crumb, Federal Reserve Bank

5.2 ARPA Network Security Project 74

Mr. Stephen T. Walker, Defense Advanced Research Projects Agency

5.3 Electronic Funds Transfer Application 80

Mr. Jack McDonnell, EFT Commission

6. Implementation and Use of the Data Encryption Standard (Picture of Participants) 83

6.1 Implementation & Use of The Data Encryption Standard within The Data Communications Environment 84

Mr. Ed. Lohse, Burroughs Corporation

6.2 Integrated System Design 94

Dr. Walter Tuchman, International Business Machines Corporation

6.3 An LSI Implementation of the Data Encryption Standard 97

Mr. Howard O. Wright, Rockwell International

6.4 A Microprocessor Controlled LSI Implementation of the Data Encryption Standard 107

Mr. Keith Warble, Motorola Inc.

7. Appendix: Question and Answer Session 116

WELCOME AND INTRODUCTION

S. Jeffery
Conference Chairman
Systems and Software Division
National Bureau of Standards
Washington, D.C. 20234

On behalf of the National Bureau of Standards, I would like to welcome each of you to the Conference on Computer Security and the Data Encryption Standard. The Conference is being sponsored by the National Bureau of Standards and the Civil Service Commission. The program that we have organized for you today has been structured to place the new Data Encryption Standard as published in Federal Information Processing Standard 46 into perspective with other measures that can be used to provide computer and data security.

The Conference has been organized into four sessions. The first addresses the major computer security aspects related to the DES. These include risk analysis, physical security and computer systems security. The second session involves those topics that should be considered in the use of data encryption. These include communications security devices, key management and system design. The third session covers the applications of the Data Encryption Standard that are presently identified. These include security projects involving encryption at the Federal Reserve network, the ARPA network and in electronic funds transfer. The final session will cover various implementations and uses of the DES. These will be discussed by members of various companies that are interested in implementing and using the DES.

In order to cover a very large subject in one day, we request that all questions be written and they will be responded to following the last session. Whenever possible, the question should be addressed to a specific speaker. Each speaker will prepare short written answers to the questions. During the question and answer period at the conclusion of the Conference, the four session chairmen will take turns reading a question and the speaker's response. The questions and answers will be published in the proceedings.

We hope that today will be beneficial to each of you and that you will find the program enjoyable.

The Data Encryption Standard in Perspective

Ruth M. Davis, Director
Institute for Computer Sciences and Technology
National Bureau of Standards
Washington, D.C. 20234

The Data Encryption Standard was approved as a Federal Information Processing Standard by the Secretary of Commerce on November 23, 1976. This Standard was developed as a part of the Computer Security Program within the Institute for Computer Sciences and Technology at the National Bureau of Standards. This paper places this standard in perspective with other computer security measures that can and should be applied to Federal computer systems either before or coincident to using the Data Encryption Standard.

NBS initiated the standards development effort leading to adoption of the DES in 1972. During this period, NBS solicited for algorithms and information upon which a standard could be based, published for comment the algorithm which best satisfied the requirements of an encryption standard, and coordinated the effort with both the potential using communities and supplying communities.

This paper outlines the environment surrounding and the history of the Data Encryption Standard and discusses the objectives of additional standards to be developed within the computer security program.

Key words: Computer security; encryption; standard.

1. Introduction

There are very few of us today whether we are computer scientists, managers, ADP facility personnel or communications specialists who have experience with encrypting and decrypting information in any operational environment. Therefore, there are very few of us who know what to expect when we first begin to use data encryption procedures. As we encounter problems or unexpected happenings there will be very few precedents we can draw upon for guidance. We should, therefore, try to

get the most from those individuals and organizations who have already stumbled and learned from their experiences. We will need to know who they are, whether they are in industry, Government or academia.

First, however, we need to remind ourselves as to why there are so many of us now concerned with data encryption when there were so few in the past. As might be anticipated, since we are still at the beginning of the first real sign of general or public interest in encryption, it is difficult to pull apart the underbrush and identify any real pathway. But let us try.

2. Who Has Been Using Encryption?

Prior to the mid-to-late 1960's almost the only use of encryption was for national security purposes. National security is still the predominant motive for data encryption. Other long-established uses of data encryption have been principally in foreign countries by inter-ministry networks, police and gendarmeries and embassy communications systems. How do we ascertain these other principal users of data encryption? Not surprisingly, we used the traditional market indicator--namely, who are the buyers of data encryption equipment sold by vendors. Here, even for U.S. vendors of cryptographic equipment, the market is principally foreign buyers.

In the United States at the present time, a very small percentage of companies use cryptographic equipment and encryption procedures. Hence, it is quite apparent that if we are to find and use available expertise and experience in cryptographic application, it will be from within the U.S. national security community, foreign organizations and governments and a very few U.S. companies.

3. Why Is Encryption More in Demand Now?

Since the late 1960's there have been a few newly emerging but important motivations other than national security for employing cryptographic equipment and procedures. Categorized in terms of technologically-induced changes they are simply that:

- Computer and communications technology have combined to encourage dramatic increases in the volume and speed of information collection and distribution.
- The principal mode for distribution of time-sensitive data is now electronic.
- Advances in electronic technology have made electronic surveillance and interception inexpensive and available to individual buyers.

- o Computer, communication and transportation technology have combined to make the geographically dispersed company or government the more common organizational entity with its management almost totally dependent on electronic means of information transmission. Categorized in terms of real or perceived threats, these new motivations for employing encryption can be put in different terms--namely, in a rough chronological order of emergent threat as follows:

    - Organized and intentional attempts to obtain economic or market information from competitive organizations in the private sector.

    - Organized and intentional attempts to obtain economic information from government agencies.

    - Inadvertent acquisition of economic or market information.

    - Inadvertent acquisition of information about individuals.

    - Intentional fraud through illegal access to computer data banks with emphasis in decreasing order of importance on acquisition of funding data, economic data, law enforcement data and data about individuals.

    - Governmental intrusion on the rights of individuals.

    - Invasion of individual rights by the Intelligence Community.

4. What Is The "Cryptographic Marketplace?"

Faced with this sporadic but increasing demand for cryptographic equipment, what kind of cryptographic marketplace exists? Obviously, the cryptographic marketplace has a very long history since equipment and procedures for transforming data into unintelligible form and then transforming it back into intelligible form have been used for thousands of years.

However, looking just at the 1970's, before the advent of the NBS data encryption standard, the cryptographic marketplace was and is large, competitive and one in which caveat emptor or "buyer beware" was the prevalent theme. There are about 150 manufacturers of discrete

cryptographic devices world-wide of which somewhat less than 100 are American companies. Most cryptographic equipment is now electronic where just a few years ago it was either mechanical or electro-mechanical. There are also a very few--probably less than five--companies world-wide that sell software encryption packages.

If you really dig in and read company brochures, you will find about a dozen major manufacturers with what we would call a full line of cryptographic equipment, e.g., equipment for data with different transmission speeds, for different types of channels and transmission methods, for off-line and on-line use, etc. My estimate is that more than 75% of these dozen companies are foreign manufacturers.

The commercial equipment is generally described in the above terminology, with additional descriptors of allowable key variations and in "working principles." We can refer to the working principle as the encryption algorithm.

As you may recall, the marketplace was described earlier as one of "buyer beware." This is because the intricacies of relating key variations and working principles to the real strength of the encryption/decryption equipment were and are virtually unknown to almost all buyers, and informed decisions as to the right type of on-line, off-line, key generation etc., which will meet buyers' security needs have been most difficult to make.

It was into this arena that the National Bureau of Standards entered in 1972.

5. Legislative And Governmental Responsibilities

Responsibilities for design, use and applications of cryptographic equipment were not clearly defined in 1972; they are still not clearly defined in 1977.

NBS, under its Brooks Act (P.L. 89-306) responsibility for setting Federal standards for effective and efficient uses of computer systems, initiated a much needed program in computer security in 1971. It pursued as an essential part of computer security the development of data encryption standards. The purpose of the NBS data encryption standards development effort was to protect computer data in transit or resident in computer systems and networks.

The primary constituency under the Brooks Act for NBS' data encryption standards were Federal agencies; the secondary constituency deriving from NBS' responsibilities as a member of the Department of Commerce was the general buyer not operating under national security provisions and directives.

Responsibilities for cryptographic R&D and use in national security activities are fairly well defined under the National Security Act of 1947 and under the amending Executive Order 11905 of February 18, 1976. Under this Executive Order, the National Security Agency serves "under the Secretary of Defense as the central communications security authority of the United States Government" and is responsible for the "conduct of research and development to meet the needs of the United States for signals intelligence and communications security." NBS has asked for and received the unique and very valuable assistance of NSA since 1972 in NBS' effort to provide data encryption standards for its constituencies.

The only recent relevant Congressional legislation is the Privacy Act of 1974, under which OMB assigned NBS responsibilities for the development of computer and data standards to meet the needs of the Act. Data security is not a requirement of the Privacy Act of 1974. However, data security is one of the means best suited for meeting requirements of the Act.

As of 1977, NBS' data encryption program and its recently issued Data Encryption Standard (DES) have not, to the best of our knowledge, decreased existing competition in the cryptographic marketplace. Indeed, at least five new hardware and/or software encryption products have entered the marketplace as a direct result of the DES.

Probably the principal change in the marketplace that can be attributed to NBS' DES is the lessening of the "buyer beware" characteristic. Anyone buying cryptographic equipment which has been validated against the DES can be assured of a specific level of data security: namely that 2^{55} attempts and the use of the method of exhaustion are required to obtain any one key for the encryption algorithm used in the DES.

6. History Of The Data Encryption Standard

As I remarked earlier, the development and history of the DES have been most interesting. NBS has been directly involved for more than five years. The active standards development effort, beginning with collection of relevant information, was initiated in 1973. We solicited for information that was available in the field of cryptography that could be used in guiding our efforts. We were looking for the technical specifications of a method of encryption which could be economically employed in a variety of computer security applications typical of our assigned constituency. We wanted this information to be publicly available so that anyone desiring to adopt the standard could do so. We wanted the method of encryption selected as a standard to be amenable to various types of equipment built by the many vendors of computer and terminal equipment. We wanted the specifications of encryption to be unambiguous so that anyone would be able to decrypt the data encrypted by anyone who also adopted the standard if he had the "key" or secret variable that had been used.

Our first solicitation, in May 1973, produced nothing that satisfied these wants. This solicitation requested "proposals for information and algorithms" that could be used in developing a standard and we got a lot of unsolicited proposals to develop encryption algorithms. It seemed that a lot of mathematicians had ideas they wished to pursue. Development of encryption algorithms is not something you do overnight, however. The algorithm that we received which had the best theoretical foundation was received scratched in pencil on a sheet of paper. It was suggested that a random stream of characters be written onto two infinite length tapes which are sent to the parties wishing to communicate. The sender should add the random stream to the message and the receiver should subtract the random stream from the message. This turns out to be the only perfect security system, but we've had difficulty finding suppliers of infinite length tapes. In addition, this system has other practical problems.

Even though we received no useful algorithms from the first solicitation, a positive step was made. Interest was shown in cryptography and a need for an encryption standard was demonstrated. In addition, when a second solicitation was made in August 1974, several algorithms were submitted. Some were too specialized: some were ineffective. One was received that showed great merit as an encryption algorithm.

7. Review Of The Data Encryption Standard

This algorithm was published for public comment in March 1975, after undergoing Government review for acceptability as a Federal standard. This is the third phase of a standards development effort: coordination and review. However, even before this was done, procedures were worked out between NBS and IBM, the developer of the algorithm, for having the rights for making, using and selling apparatus implementing the algorithm available to interested parties under the claims of certain patents held by IBM. The terms and conditions of the agreement by IBM to grant non-exclusive, royalty free licenses under these patents are spelled out in the May 13, 1975 and August 31, 1976 issues of the Official Gazette of the United States Patent and Trademark Office.

The comments received concerning the algorithm were most interesting. The most prevalent need that was apparent from the comments was for a general education in encryption. Commentors either simply wanted information on the subject or made comments showing that they did not understand the applications and requirements of encryption. This Conference was organized by NBS and CSC to satisfy this need. The comments also uncovered an important issue regarding the competitive aspects of implementing encryption in various computer architectures. This was studied long and hard by both the legal and the technical staff of the Department of Commerce. Alternative modes of employing the proposed standard were defined and evaluated, and the best ones suggested for use in various architectures. These modes can be used to provide the efficiency needed to satisfy those concerned about this issue.

The complexity and security of the algorithm were discussed in several comments. The algorithm specified in the DES is very complex. A cryptographic algorithm that provides a high level of security must be complex. In order to minimize the impact on a general purpose computer system, a hardware implementation was specified in the standard. Hardware implementations also can be validated and are nearly immune to unauthorized, undetected modification by a potential system penetrator. Software programs are susceptible to modification and are difficult, if not impossible, to validate. However, the security of the algorithm became the most controversial issue.

A standard should be acceptable to a broad range of users. It cannot, however, satisfy all possible needs of all possible users. A standard should be amenable to change when new applications or new technology evolve. It should be reviewed periodically to evaluate any need for change. The DES was developed within this framework. Some commentors felt that the security and complexity of the algorithm was not needed in their application: they wanted a simpler one. Some felt that the security was inadequate for their needs: they wanted a more complex one. They felt that the standard should satisfy all security requirements for all possible users for all time.

The matter was studied at great length by NBS. A workshop was organized to evaluate current technology and any technology in the foreseeable future which might reduce the effectiveness of the standard. Another workshop was organized to analyze the mathematical foundation of the algorithm and identify real or potential weaknesses of the algorithm. Both workshops resulted in a consensus that the DES was satisfactory for the next ten to fifteen years as a cryptographic standard. No methods for obtaining a key that you, as users, select to protect your data are known short of trying all theoretically possible keys.

There are 7.2×10^{16} possible keys for use with the DES. This means that a key would have to be tested every microsecond for the next two centuries in the fastest computers expected in the next few years. A machine consisting of a million special purpose electronic chips, each doing a test in a microsecond was suggested as a threat in the comments. Our workshop on technology concluded that such a machine, although capable of deriving one key in a day, given matched plaintext and cipher, would cost over $70 million to build between now and 1990, be 256 feet long, draw millions of watts of power, and anyone attempting such a task would have a very low probability of success. I do not want to understate the issue of security but I do want to put it into its proper perspective. The risks to data encrypted by the DES will come from sources other than brute force attacks.

Before leaving this issue, I would like to provide some special guidance. The key used with the DES is the key to security. A cliche, disgusting as it may be, is often easy to remember. No matter how good the algorithm and no matter how good the equipment, the security provided by encryption is only as good as the protection you give the key. Methods for accomplishing this will be discussed today and for many

years as systems are developed. Keys should be random; keys should be independent; keys should never have any part predetermined. Failure to follow these rules or compromises in their achievement will compromise the security equivalently.

8. Related Security Measures

Other security measures related to the use of encryption are scheduled for discussion in the next session. A risk analysis is specifically recommended in the DES before encryption is selected for use. Administrative security should be adopted before encryption is used and must be expanded to include the procedures of key handling when encryption is implemented. Physical security is always required in various degrees in all computer systems. Additional requirements for protecting encryption equipment must be satisfied when encryption is used. Finally, the technical implementation of encryption equipment must be performed for an effective cryptographic system. These areas will be discussed in depth throughout the day.

The DES was adopted as a Federal standard on November 23, 1976, and published as Federal Information Processing Standards Publication 46 on January 15, 1977. Each of you received a copy in your registration packet. The standard is divided into two sections: the announcement section and the specification section. The announcement portion gives the administrative ground rules for following the standard. Every agency is responsible for complying with the standard. Encryption should only be dictated for use from within an agency and only after an in-depth risk analysis is done. When encryption protection is required and if the data is unclassified, then encryption hardware should be procured when it complies with FIPS PUB 46 and used to provide the desired protection. The specification portion defines unambiguously the algorithm to be used to encrypt and decrypt data. Related administrative information should be obtained from the announcement portion. The effective date of the standard is July 15, 1977, and Federal agencies are to comply with the standard after that date.

In a communications application the DES does not stand alone. Existing standards must be used and additional standards are needed. Existing Federal Information Processing Standards (FIPS) and Federal Telecommunications Standards (FTS) are to be used when implementing the DES in communications. However, additional standards for the electrical, mechanical and functional aspects of stand-alone, add-on communications security equipment utilizing the DES are needed. Standards for incorporating DES devices in terminals and communications processors are needed for an effective cryptographic system.

A technical subcommittee for developing a standard for the use of DES in communications has been established by the Federal Telecommunications Standards Committee (FTSC). An ad hoc committee, under the leadership of NBS, investigated the need for such a standard. The recommendations of the ad hoc committee were adopted by the FTSC and endorsed by

the FIPS Coordinating and Advisory Committee. The recently approved formal subcommittee is drafting a standard for review and approval as a joint Federal Telecommunications and Information Processing Standard. The formation of the subcommittee was recently announced in the Federal Register. Technical contributions and comments are welcome from interested parties from both the public and private sectors.

9. Support of the Data Encryption Standard

The final topic I would like to discuss this morning is the support of the standard, the final phase in standards development. NBS will support the standard in various ways and you as potential users can obtain assistance from several sources in adopting the standard. A Data Encryption Testbed has been established within the Institute for Computer Sciences and Technology at NBS to provide some of the assistance. Two major services are being performed. First, a validation service is being established to test hardware devices for compliance with the specifications of the standard. The standard specifies a transformation of 64 input bits into 64 output bits based on a 64-bit key. It also specifies that hardware be used to perform this transformation. NBS has defined a set of tests which provide a high degree of assurance that the hardware implementation performs the transformation correctly. Vendors intending to supply such devices to Government users must have the devices validated. This service will be done by NBS on a cost reimbursable basis. The service will conform to the administrative regulations found in NBS Special Publication 250, Calibration and Test Services of the National Bureau of Standards. Agencies seeking to procure DES devices should use the wording of Federal Property Management Regulation 101-32 presently being amended by the General Services Administration. Finally, the responsibilities of the National Security Agency, formulated in Executive Order 11905 dated February 18, 1976, include assisting Federal departments and agencies in implementing communications security and determining specific security requirements in this area.

The second use of the Data Encryption Testbed is to develop and evaluate methods of using the DES in various applications. Additional standards are required for assuring compatibility among devices employing the DES in specific applications. A fundamental goal of the DES was to provide a basis of compatibility among various devices in various applications while providing a high level of security. A standard should not dictate all of its applications within the standard. Innovative implementation and application are the bases for competition in providing products or services meeting a standard. No standards effort should attempt to stifle competition or innovation. Standards should either be adaptive or amenable to change. Additional standards can be built on fundamental standards in selected applications to provide compatibility. The DES is a fundamental standard for data communications security. A Federal task group has been established to assess the need for and scope of additional standards in cryptographic systems. Information regarding

the validation tests of DES devices, as well as the standards efforts in data communications security, is available from the Systems and Software Division of the Institute for Computer Sciences and Technology at NBS.

10. Concluding Comments

In summary, the Data Encryption Standard has been a forerunner in a structured standards development process. The Federal Government took the initiative in developing a standard which satisfied its own identified need. A cooperative effort was established within the Federal Government and between the Government and private industry. For the first time, a Federal standard is publicly available that can be used to provide a high level of cryptographic protection for computer data. A very high level of public interest has been demonstrated throughout the development process. Private industry will be the suppliers of devices complying with the standard. Government agencies, as well as private organizations, will be the users of the devices and consumers of the services based on the standard.

All Federal agencies have been requested by NBS to state their needs for additional Federal information Processing Standards and to support the subsequent efforts in satisfying these needs through a cooperative standards program. Only through efforts such as these, supported by private individuals and organizations, can computers be made more effective and more secure.

Computer Security Risk Assessment

Robert H. Courtney
IBM Corporation
Systems Research Institute
291 E. 42nd Street
New York, New York 10017

The following paper has been extracted from the verbal presentation of Mr. Courtney at the February 15th Conference. A written paper had not been submitted at the time of publication of these proceedings.

1. Introduction

My objective today is to convince you that you should not spend one nickel on computer security unless you can cost-justify that nickel, that there is a way of cost-justifying that nickel, and that in all probability you should be getting on with it. A convenient way to start is by sharing with you some of the observations that were made after looking at over four hundred data processing installations. These installations had already become aware of computer security for some reason. They did not have to be made aware of the problem; they were already aware. However, for one reason or another, they had not achieved a level of security which they considered adequate.

The most probable reason for not achieving an adequate level of security is their failure to prioritize the problem. For most of us human beings, especially those who are technically oriented, we would like the problem to be technically challenging. This is a difficulty in the area of data security; the fundamental problem is not intellectually exciting.

2. Prioritized List of Computer Security Problems

I feel that there are six major problems in data security. The first major problem is simply errors and omissions. The employees committing errors or failing to perform specific acts are typically honest. They simply are not competent to perform the job adequately at all times. The dishonest people of this world will never be able to contend with the incompetent in the damage they do. The incidence of errors and omissions probably accounts for 50-80% of the data security problems I have encountered in my discussions with ADP managers. If a manager does not account for the problems in this first category, he will never be able to cost justify the security measures that he chooses to implement.

The second major category of data security problems is that of dishonest employees. It is apparent after analyzing this category that the vast majority of incidents do not deal with highly technilogical failures. For the most part they are clerks and operational people who are misusing their powers in not just doing their job, but in doing something else. Who steals from Accounts Payable? The person working in Accounts Payable. Who steals from Payroll? The person working in Payroll. The people working in Inventory do not steal from Accounts Payable; they steal from that part of the system they know best.

In the third place clearly is fire. It is not because ADP processing equipment is highly flammable. The last significant fire we had in a computer was an old IBM-650 computer in 1957. For the most part, computers burn because the fire starts in them, but they keep burning because of the flammable material around them. Most people have put their fire protection where the computer installations are rather than where the combustibles are. We seem to place our security measures where they do the least good. There is a longer lead time for obtaining pre-printed forms required for the day-to-day operation of many companies and Federal organizations than there is in the CPU that does the processing.

In a clear fourth place is the category of disgruntled employees. As opposed to dishonest employees, disgruntled employees do not have an economic motive for doing what they do. There are relatively few instances of problems caused by disgruntled employees but unfortunately the dollar value of these incidents is high. The important point here is that there is not a case known in which an employee, happy and honest on Tuesday, came into work on Wednesday and took the place apart. For the most part, the disaffection grows over a significant period of time and it is partly the insecurity or cowardliness of first level management that keeps us from catching these potential problems. Rather than meeting a problem head on, we would rather hope that it will go away. It is better to move such a person out of a sensitive position than to suffer the possible consequences.

In the fifth place is water. Floods are not the major problem in this category; broken water pipes and leaking roofs are the big probmems. One can deal with this problem primarily with a fifteen foot roll of polyethelene plastic and a pair of scissors. The higher the building or the newer the building, the higher the probability that it will suffer water damage. A fire on the 23rd floor of one building, quenched with water, knocked out a center on the 8th floor because of leaking water.

In last place are strangers. These are the people who we do not know and are not our employees. These are the people who tend to mount more technologically superior attacks against our system. As we grow toward Electronic Funds Transfer systems, we may see a higher number of incidents in this category. Given this order of prioritizing, you may be able to get a measure of the risks associated with

your particular system.

3. A Risk Analysis Methodology

This approach to risk analysis is based on a listing and evaluation of all of the data files stored and processed in a computer system. The person doing the risk analysis must then look at all of the things that can happen to those data files. There are only six "bad things" that can happen to data files. These are: accidental destruction, disclosure and modification and intentional destruction, disclosure and modification. At each intersection of "data file" and "bad thing" in a matrix, I would like to see three numbers. First, the dollar impact, very grossly stated, i.e., within an order of magnitude, of the impact of this "bad thing" happening to the data. The next number represents a probability that this "bad thing" may happen to the data. The third number in the matrix is the product of the two. This number represents an annualized risk, i.e., the number of dollars that it may cost per year. Only if I am able to come up with an annualized risk, measured in dollars, am I able to collect and apply those security measures which are cost effective. We have enough data collected from individuals performing a risk analysis to be assured that this method does work. This approach will also identify the problems which are cheaper to tolerate than they are to solve, and there are a fair number of those. There are a number of expensive security measures which will protect us against security problems which we almost never have. We must not use those.

The use of this matrix also identifies those processes or operations which a company or a Federal agency must be able to perform in order to get their job done. Most of them will actually be able to operate on an emergency basis with only 15-20% of the data processing capabilities that they normally require. However, an ADP manager must determine before hand what comprises this 15-20% of critical ADP operation. The risk analysis should yield a good indication of which processes are critical.

The National Bureau of Standards is publishing this approach to risk analysis in a document entitled "Automatic Data Processing Risk Assessment, "NBSIR 77-1228 (available as PB 265950 from the National Technical Information Services, Springfield, Virginia 22161).

Data Encryption and its Relationship to Physical Security Planning

Robert V. Jacobson
Chemical Bank
New York, New York 10041

Data encryption is a powerful tool for protecting data against discovery by an unauthorized person. However, use of data encryption does not automatically solve all security problems. The ADP security planner must examine the attacker's perception of an encryption protected system if he is to select other security measures wisely.

Key words: Encryption, data security.

1. Introduction

Over the past decade, growing emphasis has been placed on security for automatic data processing (ADP) systems for three reasons. First, ADP hardware and software are very costly. It is not unusual to have value densities of $1000 to $3000 per square foot. Second, many organizations now use computers to control daily operations. If the ADP system ceases to operate for whatever reason, the organization may suffer serious losses. Consequently, security measures to protect against damage from fire, floods, sabotage and the like have become increasingly important. Third, it is now common to find ADP systems which control valuable assets, money, goods, services or proprietary information. Most recently, we have seen great interest in protecting personal information against improper disclosure. As a result it has become important to provide effective controls over physical access to ADP resources to minimize the exposure to fraudulent tampering with data, programs, hardware and to the theft of information.

The objective of the ADP security planner is to select an array of security measures with an attractive cost/benefit ratio. That is to say that the cost of the security program is exceeded by the reduction in expected losses which the security measures are expected to bring about. He makes this selection based on the results of a risk analysis. He first forecasts the loss which each of all possible risks can be

expected to cause. (Of course, to make the process feasible, he will aggregate similar risks into a finite number of risk types e.g. major fire, minor fire, small fire, etc. and he will use simplifying assumptions and judgemental predictions.) Next, he looks at the expected losses, beginning with the largest one, and looks for security measures which can reduce the losses at a cost less than the reduction, so that there will be a net gain to the organization. This sort of analysis has led to a general emphasis on physical security measures simply because the cost/benefit ratios are more attractive than other more abstract security measures. As a rule this stems from uncertainty about the effectiveness of the more abstract measures. However, once satisfactory physical security measures have been installed, the prudent ADP manager will want to look at other measures like data encryption.

When contemplating data encryption, there are two key points that one should keep in mind. The first point is that data encryption only accomplishes one thing: it makes the discovery of the encrypted information by an unauthorized person more difficult and accidential discovery becomes extremely unlikely. It is important to note the distinction between more difficult and impossible. Bearing in mind the specific function performed by data encryption, it should be obvious that the adoption of data encryption as a security measure does not eliminate the exposure to other computer security risks.

The second point is that the management of data encryption keys will not somehow take care of itself. Explicit procedures, safeguards and audits must be adopted for the management of keys at the same time that data encryption devices are installed. Depending on circumstances these costs may not be trivial. It is not uncommon to hear that an access control device for a door only costs X-dollars. No mention is made of the costs to install and service the device, prepare and issue special I.D. cards and train personnel in its use. Without doubt the same sort of thinking can apply to data encryption. It seems likely that the cost of hardware to implement the NBS Encryption Algorithm will drop dramatically as a volume market develops. The price history of four-function pocket calculators during the period 1972-1975 provides an excellent model of a learning curve for large scale integration production costs. Therefore, the security planner must guard against the temptation to equate the total cost of data encryption with the cost of the hardware.

Given these considerations, what specifically should the computer security planner do regarding physical security as it relates to data encryption?

2. The Criminal's Viewpoint

Let us consider for a moment the problem that data encryption poses for the criminal as he attempts to gain knowledge stored or transmitted by our computer system. And let's begin by assuming that he cannot extract encrypted data without a key. We will assume that analytical extraction of a key is economically infeasible and we will assume

that trial and error extraction of a key is seen by the criminal to be more expensive than other, more conventional modes of attack. Our criminal has three choices. He can attack the unencrypted parts of the system, he can seek to compromise the encryption key by bribery or extortion or he can give up and go elsewhere. Of course, we hope he will give up but bear in mind that he will do so only if it is his lowest cost option. If the desired information is not available elsewhere and the cost of failure is greater than the cost of success, we should assume that he will continue his efforts.* Hence, we should first consider not how valuable the information is to us or how great our loss would be if it were improperly disclosed but rather what reward the criminal gets for stealing it and where else he might go to get the same information.

Crime prevention specialists often speak of crime displacement. If we double the foot patrol in the ninth precinct we can cut street crime in half. But can we? Sadly, we find that crime in the neighboring precincts has increased almost proportionately to the decrease in the ninth precinct. Therefore, if our analysis of the criminal's perception of his reward-to-risk-ratio suggests that he will not choose to go elsewhere, we should assume that we have only displaced, not eliminated, the crime. If it appears that data encryption will only displace the attack to some other part of the ADP system, it would seem as though data encryption were of no value.

Of course, this is not the case. The objective of the security planner is to use each security dollar he spends to prevent as many loss dollars as possible. The way he reduces crime losses is by making the reward/risk ratio less attractive to a prospective criminal. Since the operative reward/risk ratio is the one which applies to the most weakly defended part of the system, strengthening that part of the system will, in fact, reduce crime losses. In the ideal case, all parts of the system would be perceived by the potential criminal as having the same reward/risk ratio.

3. Analysis of a Typical Case

Figure one shows a specific example of these considerations. Assume that the criminal wants to see the transactions performed by the operator at the remote terminal. He has five reasonable possibilities: (1) subvert the remote terminal operator; (2) obtain the information at the remote terminal without the operator's knowledge from emanations, discarded printout, hidden camera, etc.; (3) tap the data circuit to the terminal; (4) subvert the console operator and get from him a printout of the transaction journal for the terminal; (5) obtain possession of the transaction journal medium. (There are, of course, other more remote

*In this regard it is important to understand that generally speaking the rational criminal is motivated by his perception of the ratio of his reward to his costs, risk of discovery and punishment, out of pocket expense, etc. regardless of the amount of the victim's loss.

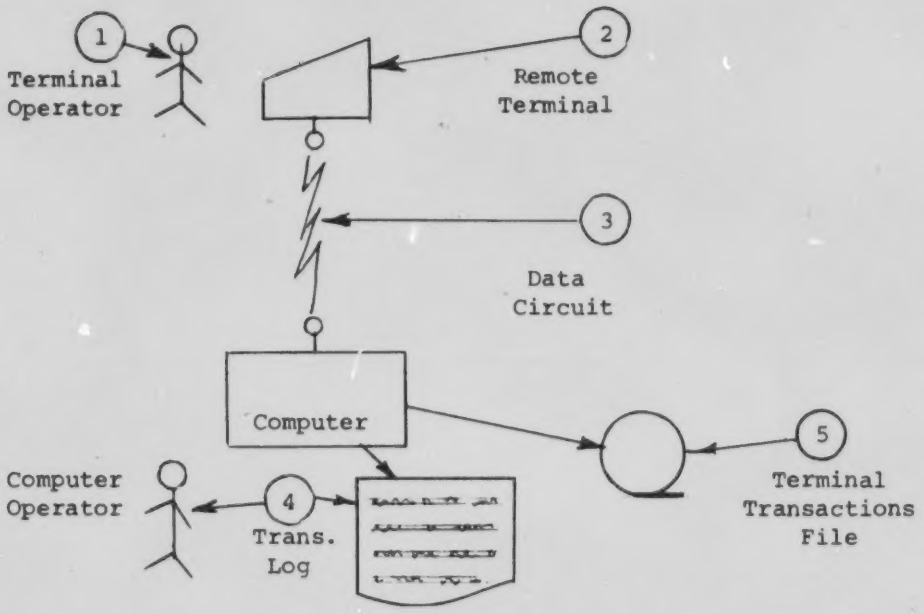


Figure One: A representative data communications system showing five points of attack.

possibilities but these five are adequate to illustrate the point at hand.) Figure two shows how the criminal is likely to evaluate each of these possibilities with and without data encryption of the data circuit and the terminal transaction file. The evaluations placed on each of the factors are the author's and the reader may not always agree but he will probably agree with the overall conclusion. An attack on the data circuit is by far the most attractive in most instances. The skillful criminal with adequate resources can arrange the tap so that once it is in place, there is no traceable link between the tap and the criminal's base of operations. As a result, the danger of discovery and punishment is much reduced compared with attacks on other elements where he must physically enter protected areas. He will get exactly the information he wants and he will get it in real time which may be important in some cases.

4. Physical Security Requirements

How will the criminal respond if he finds the circuit protected by the NBS Encryption algorithm? Assuming the algorithm to be uneconomic to crack, he only has two choices; get the key or attack elsewhere. Figure two suggests that he will go after the terminal unless he believes he can get the key itself.

The physical security requirements can now be seen to be these:

1.) The protection of the remote terminal against snooping must

Target	Real Time?	Danger of Discovery	Probable Success		Preference**	
			No D-E	With D-E	No D-E	With D-E
1. Terminal operator	yes	high	high	high	4	2
2. Terminal	maybe	fairly low	very low	very low	2	1
3. Data circuit	yes	low	high	zero*	1	4
4. Computer operator	no	high	fairly low	fairly low	5	3
5. Transaction journal file	no	fairly low	low	zero*	3	5

* Assumes data circuit and journal file are encrypted and keys are not compromised.

** Probable order of preference perceived by the potential criminal, 1 being most preferred and 5 least preferred.

D-E = data encryption

Figure Two: The criminal's evaluation of alternate targets.

be strong enough to deflect the criminal. There must be no easy way for him to tap into the terminal where text is in the clear, to pick-up electromagnetic or accoustic emanations from the terminal, to get copies of printout or to place a TV camera to observe the screen and keyboard. Obviously, he should not be able to get the key from the terminal itself.

2.) Similar measures must be taken at any point in the computer facility where the terminal transactions can be displayed or intercepted in clear-text.

3.) Methods used to generate keys, carry them to the terminal and to install them must be proof against undetected compromise.

4.) Access to the computer and the terminal must be limited to the least number of individuals and all such access must be a matter of record.

The last point seems obvious but may have implications not immediately apparent. Let us assume that we have installed highly secure data encryption and such effective physical security that there is no direct way to get the desired information. At this point our criminal might very well seek to install his "tap" inside the computer. Rather than approach

the terminal operator or computer room personnel, both obvious targets, he might try a more indirect route. Without revealing his real objective, he might try to patch the computer's control program to allow him to eavesdrop on the remote terminal or the key generation process. How do we stop this? Only by complete control over all changes to hardware, system software and applications programs. Obviously change controls are meaningless if they are not implemented with credible controls over physical access to system resources.

5. Evaluating Data Encryption

A natural reaction at this point is to question the value of data encryption. It seems only to have forced a lot more security measures on us. Of course, that is not the point. What we have done is to identify all the points at which the information is exposed to criminal attack and tried to make all the points equally difficult to attack. Failing that we will simply displace the crime from the weakest to the next weakest point, perhaps at little added cost to the criminal.

What we must do is to evaluate data encryption as a security measure in terms of the kinds of losses it can reduce and in comparison with other measures which achieve the same loss reductions. We recognize that data encryption protects against losses resulting from unauthorized disclosure of information but nothing more and only protects at the points where it is used. Consequently, we won't expect data encryption to solve any other security problems. We can also see that there are other ways to protect the data circuit and transaction journal in our example. The journal medium (tape reel or disk pack) could be removed from the ADP hardware by a two man team and kept in a safe with two combination locks. Likewise, we could use special pressurized coaxial cable for the data circuit which alarms if an attempt is made to cut through the jacket. As a back-up, special electronics could measure the characteristics of the data circuit and detect the slight electrical changes caused by a tap. The reader probably can imagine additional measures. We can estimate the cost and probable effectiveness of each of these potential measures with some confidence.

The reason we are interested in data encryption is that in most cases it will be much cheaper than any other potential security measure. Once, data encryption has been identified as the most economical security measure, we should consider the relative merits of hardware and software implementations. Cost differences will depend on particular circumstances but hardware has a number of advantages. Fraudulent alteration of the algorithm is much more difficult and when LSI is used it will be substantially impossible. With well designed hardware, the key will "evaporate" if power is turned off or the container is opened. In extremely critical applications, the container can be equipped with devices to sense tampering and signal key erasure. The security auditor will certainly prefer these features since they are all auditable.

This leads us to a final point. We can never be sure that our defenses will work as expected or that we have correctly anticipated

how the potential criminal will attempt to attack our system. For both reasons it is important to have an effective audit program which operates unpredictably in both time and space. Both the criminal and in-house personnel who might be his targets should be able to predict neither when a given function or area will next be audited nor how the examination will be conducted. The credible audit program will decrease the criminal's assurance that he will not be caught and so further aid in detering the crime. This is particularly significant when the criminal would, except for the audit program, predict a zero probability of discovery. Thus, even though he knows that an area key to his planned crime has never been audited, if he knows that it might be audited tomorrow he will think twice before going ahead.

In summary, data encryption will provide a very high level of protection for data but other points at which the data are exposed must have commensurate levels of protection if we are to enjoy the full benefits of data encryption. Security measures used during generation, distribution and installation of encryption keys should be strong enough to discourage attack. All security measures should be supported by a high quality audit program with an unpredictable schedule and scope. Finally, management must recognize the need to analyze all security needs in terms of both risk and loss exposures and to strive toward a balanced, economically sound security program.

Computer Systems Security and the NBS-DES (Beyond Line Encryption)

Clark Weissman
System Development Corporation
2500 Colorado Avenue
Santa Monica, California 90406

The recent adoption of the Data Encryption Standard (DES) by the National Bureau of Standards has created significant interest in the area of cryptography. There are numerous considerations to be made when designing a cryptographic system. The NBS-DES must be embodied in a system employing automatic, down-line key management and end-to-end encryption to be truly effective in a computer network. This paper reviews several issues in this area and suggests solutions.

Key words: Cryptography, end-to-end encryption, key management.

INTRODUCTION

Encryption can do more than protect data in transit. It can be employed to enhance the security of computer systems as well as to authenticate users, grant them access to system resources, and dynamically enforce that authorized access. The NBS-DES is an excellent vehicle for achieving these ends when used in a system-wide manner that employs automatic, down-line key management and end-to-end encryption.

System Development Corporation is involved in developing these hardware and software encryption techniques and practically applying them to improve the computer security of Electronic Funds Transfer Systems (EFTS). In this paper I review the security issues and suggest solutions.

PROTECTION: A SYSTEMS PROBLEM

Security is a "weak link" phenomenon with exposure arising from high-valued assets leaking from a flawed information system through the planned efforts of exploitative criminal interests. The information system consists of entry/display, delivery, and processing subsystems which depend on vulnerable computers and software. Countermeasures must be balanced to raise the protection of "weak links" in a uniform manner by application of a system-wide plan.

The plan elements shown in figure 1 are: (1) a protection policy, reflected in the requirements for the system; (2) omnipresent enforcement of that policy by the total hardware, software, and people components of the system; and (3) trustworthiness accreditation of the system at each stage of its lifecycle development. Let us look at each plan element in turn.

POLICY PROTECTION REQUIREMENTS

Protection policy requirements may be geared to counter threats from different sources. Safety requirements counter failure or accidental exposure of sensitive data. Countermeasures are based upon using trusted components, component redundancy, and trouble-detection and backup procedures.

Privacy requirements deal with constraints placed on authorized users who disclose data inadvertently or by exceeding their authority. Countermeasures depend on increasing the granularity and control of information. If users and data

THE CORNER STONE OF A SECURITY PLAN

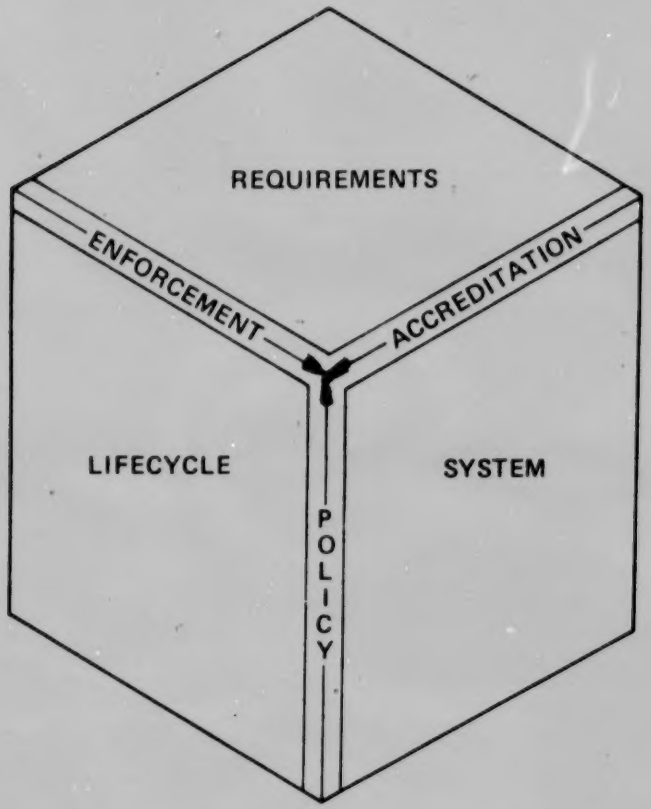


Figure 1

are explicitly identified and differentiated in terms of data-item sensitivity labeling and levels of user authority, control can be imposed to limit access to the least amount of privilege necessary to accomplish a job. Also, the improved granularity can enable fine-grain transaction journaling and accounting for authorization checks.

Lastly, security requirements address the sophisticated planned penetration attack on the system to steal data or sabotage the system. Figure 2 summarizes these threats and countermeasures. Most noteworthy is that human intelligence can seek out or plant system hardware or software flaws to achieve these security violations.

LIFECYCLE ACCREDITATION

Once the protection policy is defined, the resulting requirements must be satisfied by the enforcement system. Trustworthy enforcement options increase with the lead time available before delivery of the system, from future research, through new system developments, to operations on existing systems. Figure 3 summarizes the state of the art: only future systems show promise of solving the security problem. However, privacy and safety requirements can be addressed today with procedural and physical barriers, and with some new design retrofit.

In future systems, the security policy must be enforced by the total information system in the entry/display, network delivery, and CPU processing subsystem elements. Since each of these involve computers and software, they are all vulnerable to common generic problems. However, the specific nature of the tasks for the entry and delivery subsystems makes the use of encryption quite attractive. Furthermore, the central processing subsystem plays the important support role of ensuring the security integrity of the other subsystems.

PROTECTION ENFORCEMENT IN THE ENTRY/DISPLAY SUBSYSTEM

User and system authentication is the principal enforcement function addressed by the entry/display subsystem. Threats and counter measures are described below according to the increasing virulence of the threat.

Impersonating someone is the simplest threat, effective on systems without mandatory ID checks. These checks should be made based on a unique Personal Identification Number (PIN) manually entered on the terminal. It is best for the PIN to be committed to memory and otherwise carefully protected,

PROTECTION POLICY: SECURITY REQUIREMENT

THREAT

1. ASSET THEFT, FALSIFICATION, SABOTAGE
2. FLAW FINDING
   - STRESS SYSTEM LIMITS, PROHIBITIONS
   - PLAN TRAP DOOR, TROJANHORSE
   - MODIFY SYSTEM CODE
   - PROCESS TO PROCESS SIGNALING
3. FLAW EXPLOITATION
   - BY-PASS OR DISABLE CHECKS, AUDITS, RECORDING
   - FALSIFY PARAMETERS
   - IMPERSONATION USER
   - PIGGY-BACK DATA COPY
   - COMPONENT SUBSTITUTION
   - OPERATOR, USER SPOOF

COUNTERMEASURE

1. SUBJECT AND OBJECT DEFINITION
2. PROCESS (SUBJECT) ENCAPSULATION (SECURITY PERIMETER)
3. SUBJECT/OBJECT ACCESS RULES (CONTROLLED SHARING)
4. ACCESS CONTROL MECHANISM (ACM)
5. SELF PROTECTION
   - ACM ALWAYS INVOKED
   - ACM OBEYS POLICY
   - TRUSTWORTHY

Figure 2

HOW CAN THE SECURITY PROBLEM BE SOLVED?

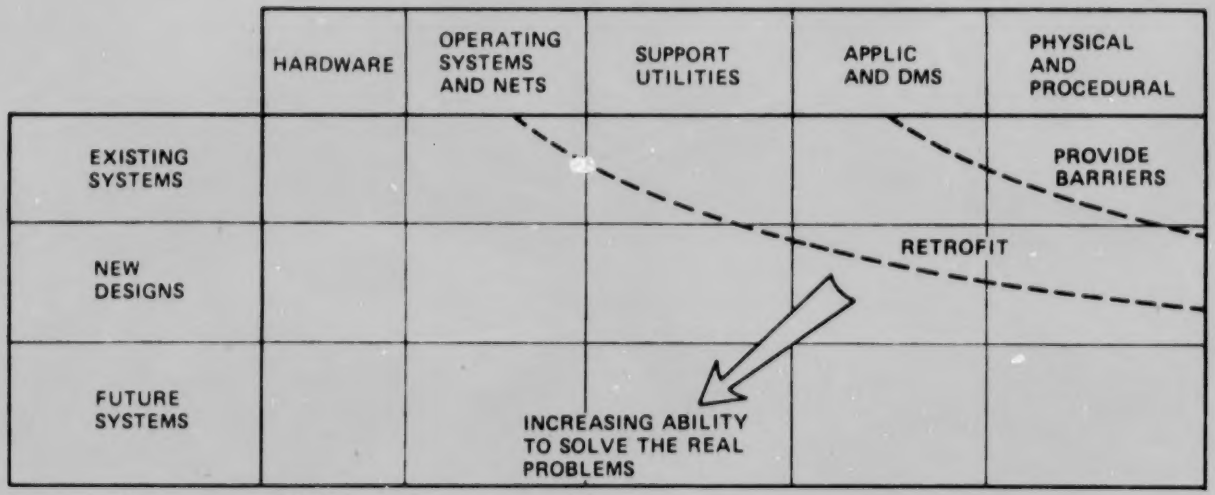


Figure 3

since it is the basic security authenticator of a user's identification credentials. The user's ID (not the PIN), a Personal Account Number (PAN), and the Cryptographic Check Digits (CCD)--an encrypted form of the PIN--can be written on a plastic card which is readable by the terminal. Lost or stolen credentials (cards) can be countered by positive ID authentication of the PIN against the CCD at the terminal, if it has the "smarts" and the host processor is offline, or downstream at the host processing subsystemm if it is online. Auxiliary checks are necessary to protect against altered or counterfeit credentials. These include secondary credentials, e.g., credit cards and drivers' licenses, or online CCD/PIN check at the host processor.

An unusual, but simple form of fraud involves spoofing a user to surrender his PIN with simulated system messages from a counterfeit system. The best protection approach is knowledgeable, alert users who authenticate the system on the other end of the dialog based on a prearranged "handshake" of randomly selected data from a user-exclusive data base.

Finally, the terminal itself may be stolen and counterfeited, permitting PIN capture and storage for later playback. Physical protection is necessary, including tamper and disconnect detection and alarm. Logical protection is possible and desirable employing the NBS-DES. DES-keys can be automatically erased upon terminal disconnect, and terminal ID's can be encrypted in transaction messages to thwart bogus message originators.

PROTECTION ENFORCEMENT IN THE DELIVERY SUBSYSTEM

Data exposure from theft of data in transit on the communications line is an old threat solved by line encryption. However, the increased use of digital traffic has led to new network architectures using security-vulnerable store and forward switches, packet processors, communication front ends, and value-added network (VAN) processors. The old line-tap threat is compounded by misrouting of messages, data leakage and theft, or message modification in these intermediate computers of the delivery subsystem. Furthermore, simple line encryption is insufficient protection as cleartext flows through these intermediate computers to permit them to perform their routing and value-added tasks. The solution is to separate message text from control text, encrypting the former (from originator to destination) and having the latter in cleartext within the delivery subsystem computers. This concept of End-to-End Encryption (E^3) counters the new network

threats but requires new hardware and systems technology to perform the "smart" selective text processing.

A number of other threats can be addressed by NBS-DES and E^3 technology. Encrypted messages may be copied off the line and altered or duplicated for later playback. Protection measures include full message text encryption with the text containing both message sequence numbers and redundancy codes. The NBS-DES is cryptanalytically sound to resist code-breaking threats. Thus, new threats will arise from operational and management employment of the NBS-DES and E^3 techniques. For example, separately evolving networks will at some future time need to exchange data. Incompatible encryption algorithms or key management schemes will restrict such system interchange. The NBS-DES is fully reversible, and all employment schemes should maintain that feature. Counter arguments advocating irreversibility based on fear of key loss and theft which can compromise data, have merit; however, those fears are best addressed by frequent, automated key change.

Frequent key changes (weekly, daily, or as needed) limit the useful key life for the thief, but only if such frequent key handling does not itself expose the keys. Frequent manual key change is both a security vulnerability and a high-cost key-management operation for any moderate sized network. Less frequent key change increases key life and theft exposure. The solution is automated key management based upon a secure, hearty protocol for loading keys downline through the delivery subsystem itself. One scheme, explored by SDC for the National Bureau of Standards, is the use of smart Network Cryptographic Devices (NCDs) controlled by a Network Security Center (NSC) as shown in figure 4 and described in the following section.

THE NETWORK SECURITY CENTER

The NSC is connected to the network, like other hosts, via a smart encryption device, i.e., an NCD. The NSC maintains a security access control data base consisting of users (subjects) and network resources (objects), and the access authorizations of each to the others. During operation, a user terminal or host calls the NSC via an omnipresent clear channel and requests an authorized connection to another resource (e.g., terminal or host). The NSC consults its access data base to validate the authorization for the connection. Since the data base is on-line to the NSC, cautious and controlled data base changes can permit revokable access authorization.

NETWORK SECURITY CENTER (NSC)
WITH NETWORK CRYTOGRAPHIC DEVICES (NCD)

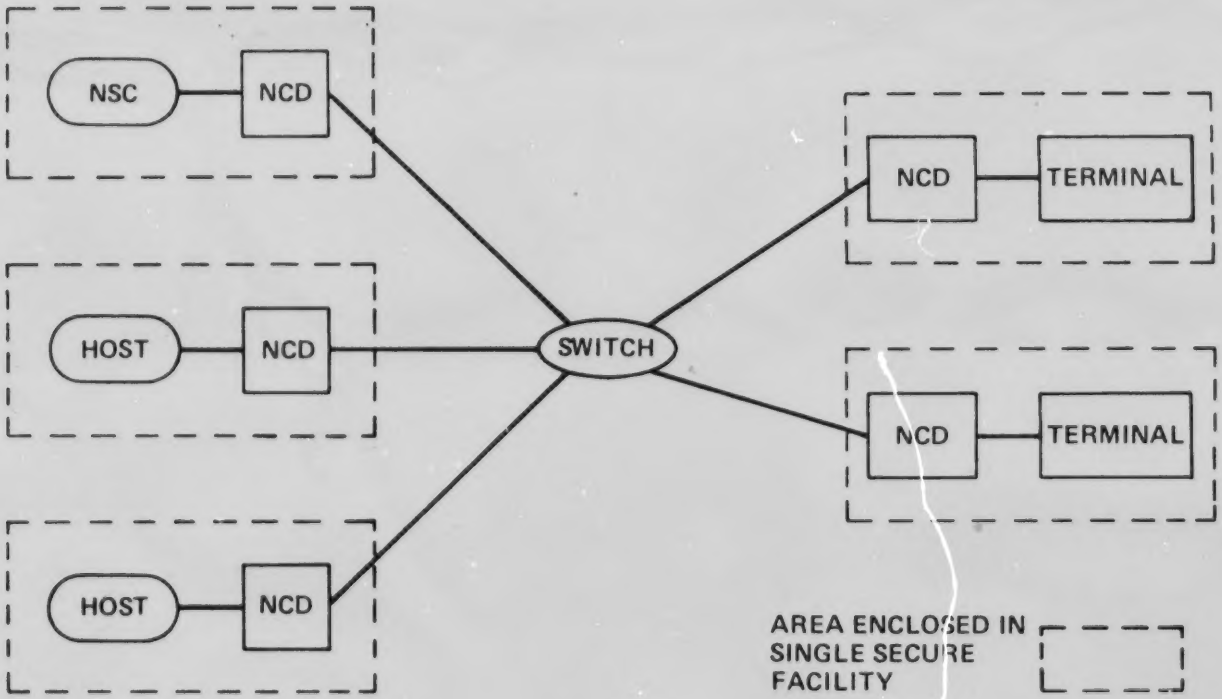


Figure 4

The NSC establishes the connection "logically" with positive action by distributing a "connection" key to the NCDs of the authorized and requesting parties. This procedure allows secure, dynamic, creation and termination of authorized connections. It permits site mobility of personnel, since their "clearances" are centrally stored and accessed at the NSC. It offers central net-wide access auditing. Also, interoperability to other nets is possible, with the NSC acting as "gateway" to the other nets or to another NSC. The NSC approach offers logically separate subnets to share network facilities and costs, thereby yielding improved security at competitive costs.

PROTECTION ENFORCEMENT IN THE PROCESSING SUBSYSTEM

Protecting the CPU and its software for shared use is the most difficult security problem. Numerous penetration studies and system software audits of current commercial operating systems have established without doubt their vulnerability to intentional, intelligent attack. Hence, those considering shared processor use among multiple applications must proceed cautiously. The best current countermeasure is not to share, but to dedicate the system to a single application within physical and personnel barriers. Another, less extreme measure, but one with less security, is to prohibit concurrent transaction-oriented applications use and software development on the same machine. Then, the only threat is from and between the application users who are constrained from generating programs to attack the system by the language of the transaction processor. It is already established that the data management application system cannot give better security than the operating system under which it operates, but it can provide finer control granularity of data objects of interest. Of particular interest, here, is the processing subsystem support to the entry and delivery subsystems security.

PROCESSING SUBSYSTEM SECURITY SUPPORT FUNCTIONS

A user enters his PIN and ID credentials at the terminal. The most secure authentication of the ID is by the on-line host which checks the PIN against the CCD for the designated user account. Offline checks at other than the host expose the system to organized fraud that counterfeits cards and PINs simultaneously. The host processor is also necessary to detect duplicate, missing, or altered messages by checking message sequence and redundancy-check numbers. Furthermore, the host must see to it that transactions are securely and

positively acknowledged. Of course, the host must log all transactions and process the log for human audit and analysis. Finally, software source data and code stored off line must be protected from accidental, or from intentional, but nevertheless unauthorized, modification. This requires good source data/code management and configuration tools, which are best satisfied for large systems by the host computer itself.

SUMMARY

In figure 5 we return to our security plan cornerstone, now complete with second-level detail. Threats to computer system security are to the system assets by exploitation of system flaws. All security countermeasure strategies aim to reduce the threats by eliminating assets (e.g., data encryption), eliminating exploiters (e.g., background investigations, bonding), and/or repairing weakness (e.g., new design). New designs are now possible with the NBS-DES, NSC, and NCD, which employ the new techniques of DES key distribution and end-to-end encryption.

CORNERSTONE OF A SECURITY PLAN

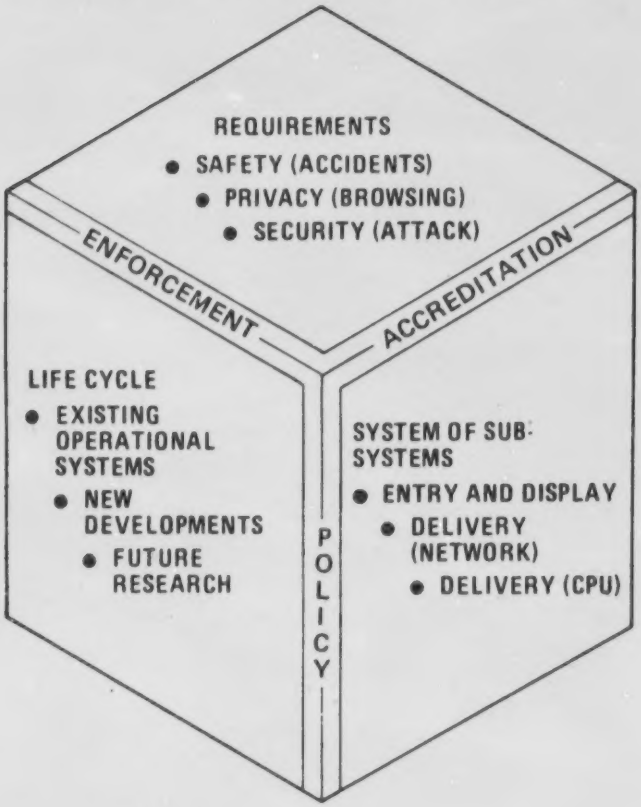


 Figure 5

Considerations in Applying an Encryption Device to a Communications Network

Barrie Morgan
Datotek, Inc.
13740 Midway Road
Dallas, Texas 75240

This paper outlines the basic considerations which must be met in applying a data encryption device to a communications network. Although the following information applies to most enciphering devices, the DES algorithm does have several unique features which merit special attention.

Key words: Cipher feedback; codebook form; forbidden characters.

1. Introduction

The recent adoption of the Data Encryption Standard (DES) by the National Bureau of Standards has spurred many potential users and suppliers of data encryption devices to investigate the application of this Standard. As previously discovered by many engineers and cryptographers knowledgeable in the area of secure communications, there are numerous considerations to evaluate when applying an enciphering device to a communications network. These considerations apply not only to the DES but to data encipherment in general. The DES (basically a block cipher) presents several unique problems when applied to a network which may or may not be block oriented.

All the parameters which must be appraised in securing a communications network are too numerous to cover in detail; however, some of the more prominent parameters are:

Initialization

Suppression of forbidden characters in the ciphertext

Synchronization

Error rate and recovery.

2. Properties of a Block Cipher

The DES, as adopted, describes a mechanism by which 64 bits of input data are operated on by a complex iterative algorithm to produce 64 bits of cipher. In the case of computer file enciphering, it can be seen that this algorithm works quite well. For example, assume one wished to encipher a file consisting of 64-bit words as shown in figure 1. Each word is pulled from the file, enciphered by the DES and replaced. Notice that each word of the file is a separate entity and can be enciphered one at a time in any order. The fact that each file word is handled separately provides the user with a great deal of flexibility. Enciphering can occur in sections. The enciphered file words can be rearranged, and portions of the file can be deleted with no effect on the subsequent deciphering.

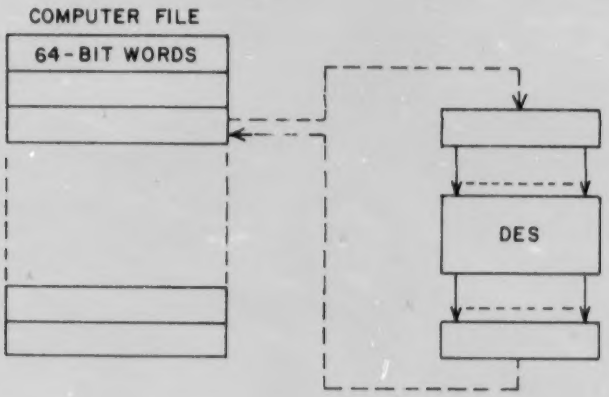


Figure 1.

When used in this manner, the block cipher requires no special initialization or synchronization. Errors are unlikely in such a local process, and forbidden characters (illegal combinations of bits) normally do not present a problem in a data file. Therefore, the block ciphering method seems to be an ideal approach for protecting a computer file. The problems appear when the secure file is transmitted.

3. Considerations in Secure Data Transmission

Let us assume that the problem is not merely to encipher a sensitive file but to transmit it via a computer switch from location A to location B and to protect it from unauthorized eavesdroppers during transmission. Normally the words are concatenated to form a serial bit stream which is then

embedded in a format required by the protocol of the switching computer. In figure 2, a simplified diagram is shown utilizing a Key Generator (KG) in the conventional method of enciphering such a bit stream. The data bit stream is presented to the modulo-2 adder simultaneously with a pseudo-random bit stream produced by the KG. Each data bit and key

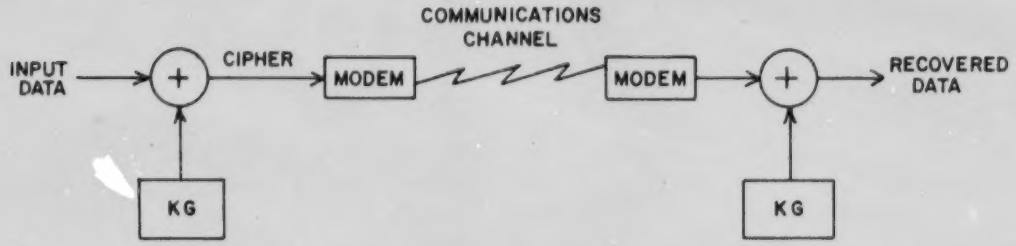


Figure 2.

bit produces a cipher bit which is transmitted via the communications channel. At the receiving device, the inverse process occurs. The incoming cipher stream is modulo-2 added with the identical key stream (identical to the key stream used to encipher the data) and the original data is reproduced.

Next let us consider the format of the message shown in figure 3. Prior to transmitting the secure data file, a header must be transmitted which instructs the computer switch as to the proper routing of the message. This part of the message must remain clear (unenciphered). An indicator denoting the beginning of the data file or the start-of-text (STX) is used by the KG to start the enciphering process.

Figure 3.

To synchronize the two key generators cryptographically, a starting point must be identified by both the receiving and the transmitting devices. This is usually accomplished by letting the transmitting device generate a random starting point. This random start is transmitted to the receiving

end to enable the two key generators to begin at the same random point. This provides additional security to the system. The random starting point guarantees that identical messages enciphered with the same key variables always produce different cipher. This is particularly important if the messages are highly formatted and are similar in content.

After the starting point is established, the KG's are stepped in synchronism as some function of the data, or of the modems. Finally, the end-of-text (ETX) halts the enciphering process and allows the trailer of the message to be transmitted in the clear.

From this one example, most of the basic problems of adding encryption to a communications channel can be illustrated. (The following summary applies to data encryption in general and not to the DES specifically.)

3.1 Initialization

The header of the message must be clear for proper computer routing and the enciphering is initiated by the STX character. The random starting address completes the required initialization of the KG's.

3.2 Forbidden Characters

Control characters normally are reserved for control of the communications channel. Therefore, it is required that control characters such as STX, ETX, etc. be transmitted in the clear. Conversely, no control characters should appear in the enciphered text. The occurrence of these control characters in the cipher could cause spurious and erratic operation of the channel and the computer switch.

3.3 Synchronization

Once the key generators are started, they must be incremented or stepped under control of the data or by the modem depending upon the type of transmission. Normally, the data start-bit is used to step the KG's in asynchronous channels. In synchronous channels, the modem clock provides the stepping signal. In either case, if a character or a bit is dropped during transmission so that the KG's lose synchronism, the remainder of the message will be indecipherable. When this happens, the ETX will not be recognized by the receiving device and the KG will not be switched off. Some recovery procedure must be initiated to start the transmission again.

3.4 Error Rate and Recovery

A single bit error in the cipher occurring during transmission will cause a single error in the deciphered data.

However, if a bit is dropped (or added), causing the two KG's to get out of step, the rest of the message will be lost. It is essential to activate a recovery procedure, usually a time-out, then restart the transmission.

4. DES in Codebook Form

Codebook form refers to the DES as published in the Federal Register in that a 64-bit data word is applied and a 64-bit cipher word is produced. When operated in this fashion the DES is somewhat analogous to a large look-up table or code book. If the same 64-bit word is applied repeatedly to the input, the same cipher is produced. This will continue until the key variables are changed.

Most of the basic problems mentioned above remain when the DES is used in the codebook form. Notice in figure 4 that the KG has been replaced with the DES algorithm. A 64-bit register has been added to accumulate the serial data bits and presents them to the DES in parallel for enciphering.

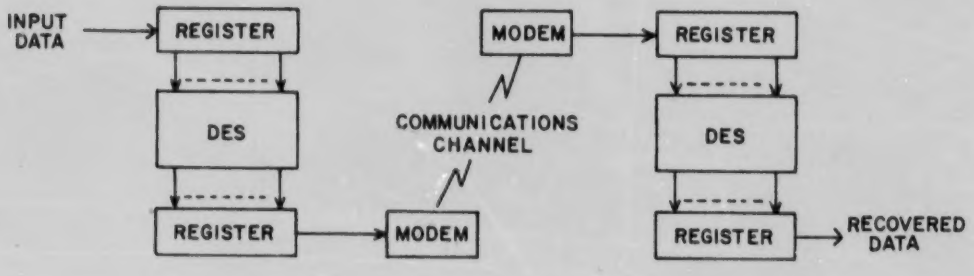


Figure 4.

The requirement for initialization still exists since the header and the trailer must remain in the clear. However, the random starting address used by the conventional KG approach is not meaningful when using the DES because each 64-bit block is a separate entity. This is one of the characteristics of the DES, which should be considered by the user. The cipher produced is solely a function of the 64 bits presented and does not depend on previous blocks. Messages which are highly formatted such as Electronic Funds Transfer (EFT) will produce the same cipher if the same input is applied. This may produce "recognizable cipher" in certain portions of the message which may not be acceptable from the security viewpoint.

The forbidden character problem is still present and is complicated by the fact that the cipher being produced in blocks is not necessarily character oriented.

Synchronization still remains critical. If a bit is dropped during transmission, the remainder of the message will be lost since the receiver will be operating on the wrong 64-bit block. The same type of error recovery procedures described above will be required. A single bit error now generates a 64-bit burst error, because the algorithm operates on each 64 bits as a block. A single bit error in the block produces a deciphered block which has little resemblance to the original.

A final problem in applying the DES occurs when the message is not an even multiple of 64 bits in length. The controller must recognize this situation and provide enough fill bits to complete the block.

5. DES in Cipher Feedback Mode

Some of the shortcomings of the codebook approach can be overcome by using the algorithm in an entirely different configuration. Figure 5 illustrates the cipher feedback mode of the DES. Here the DES is used (more or less) as a key generator. The output of the DES is modulo-2 added to the data to produce ciphertext. In this case, the key and data are added character serial/bit parallel. The ciphertext produced is transmitted and at the same time loaded into the input register which supplies the 64-bit input to the DES. The previous contents of the input register are shifted eight bits to the right prior to loading the new ciphertext character. The DES now executes another cycle and uses the first eight bits of the output to encipher the next eight bits of data. The other 56 bits are discarded. This process continues until the input register has been completely loaded with ciphertext. Notice that the receiving device is connected differently in that the ciphertext is fed directly into the input register. As soon as the input registers in

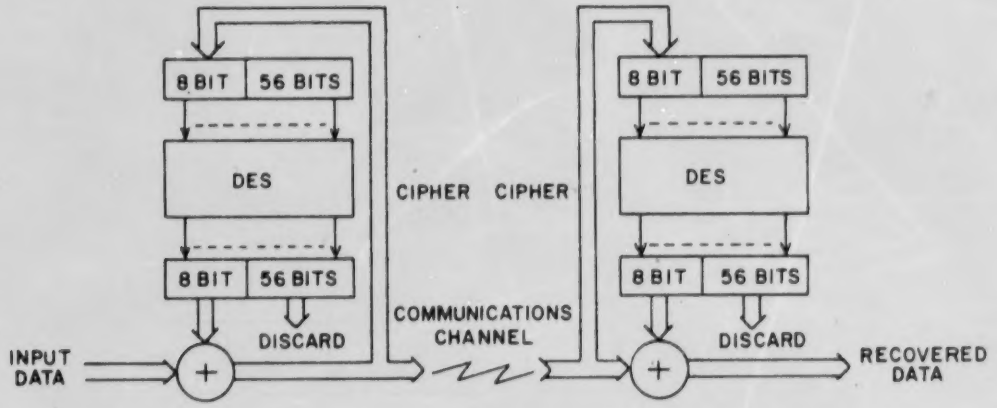


Figure 5.

both the transmitting and the receiving devices have received 64 bits of ciphertext, they will start producing identical output. Enciphering continues in this manner on a character-by-character basis.

What advantage does this configuration offer? The most obvious advantage is that the two units are self-synchronizing. All that is required for identical output is identical content in the input register. Since the input register in each device is being loaded with the same ciphertext, the output of each DES is the same. Should a bit be dropped during transmission, the receiving unit will generate invalid output until the input register has been properly filled. Therefore, we see that the loss of a bit does not cause the remainder of the message to be lost, but only a 64-bit burst error generated. Unfortunately, the DES reacts to a single bit error in exactly the same manner. In other words, each single bit error produces a 64-bit burst error. This is referred to as the "error multiplier" or "error extension" of the system.

To initialize the cipher feedback mode, the message must be preceded with eight dummy (preferably random) fill characters. The message format using the cipher feedback mode may appear as shown in figure 6. Again, the clear header is necessary for computer switching, and the STX character can be used to start the enciphering process. The random fill guarantees that the input register has sufficient data to start generating valid key. These eight fill characters can be ignored or discarded by the receiving device.

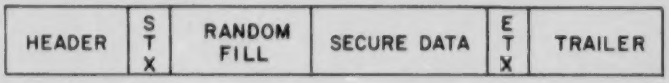


Figure 6.

The forbidden character problem is still present, but since the ciphertext is being generated on a character-by-character basis, additional circuitry can be included to suppress unwanted cipher characters before transmission.

This configuraton offers the distinct advantage of being self-synchronizing at the expense of loss in potential throughput. On high speed circuits, the maximum throughput of the DES may become more critical in cipher feedback mode. Since each enciphering cycle of the DES produces only eight bits of cipher instead of 64 bits of cipher as in the code-book configuration, higher speed is required to produce a given data rate.

6. Conclusion

The DES algorithm approved by the NBS is a significant step toward standardizing the encryption of data transmitted over communication channels. However, the algorithm itself is only one of the requirements needed to implement a secure data system. Although the Standard as adopted is readily adaptable to enciphering file data, numerous variables and options remain as to how the DES is to be applied to a switched network. The cipher feedback mode does make the DES more readily adaptable to the telecommunications environment. However, more standards must be adopted before totally compatible networks are ensured.

The Management of Encryption Keys

David J. Sykes
Honeywell Information Systems, Inc.
P. O. Box 6000
Phoenix, AZ 85005

In a system where the details of the encryption algorithm are publicly known, the overall security of the system is heavily dependent on the security of the keys. This paper discusses the various aspects of key management such as key generation, key storage, key distribution and key loading. Techniques to perform these functions are described with emphasis on data communications applications. Rather than recommend a general solution to the key management problem, numerous factors are presented for consideration by the system planner. The need for a trade-off between complexity and practicality in a real world environment is stressed.

Key words: Encryption Keys; Key Distribution; Key Generation; Key Loading; Key Storage.

1. Introduction

The NBS algorithm is based on a 64 bit key. The key can exist physically in the form of manual switch settings, a series of bits stored in a memory, holes in a punched card or bits recorded on a magnetic stripe card like a credit card. Of the 64 bits, 8 are parity bits and as such are determined by the other 56 bits. The number of possible keys is 2^{56} or approximately 7.2×10^{16}. The strength of the NBS algorithm is based on the large number of possible keys combined with a non-linear enciphering process. A good key management system must therefore make proper use of the very large number of keys available.

Now that the NBS encryption algorithm has been adopted, and several devices based on it are appearing in the marketplace, the subject of key management becomes very important. If we assume the adversary knows all about the algorithm, its implementation in your system, your operating procedures, the knowledge of the keys is the only critical thing he does not have. It has been accepted that the determination of the key by trial and error is not economically

feasible, and consequently the criminal must resort to methods of directly obtaining the key. This paper addresses the methods of safeguarding the keys during their generation, storage, distribution, loading and handling.

2. General Principles

There are no standard methods for implementing key management. Each organization must plan and implement its own system based on the particular risks and consequences of a key being discovered and used by an unauthorized person. It should be assumed that there is collusion between a person inside the organization and a person on the outside.

A quantitative assessment should be made and a key management scheme tailored accordingly. In particular, the differences should be recognized between a communications application where the keys can be changed frequently, and a media encryption scheme where the keys need secure storage during the valuable life of the data.

Whereas the encryption algorithm and its implementation details will be publicly available, all aspects of key management should be kept "secret" within the organization. Only the minimum number of trusted employees should be involved in key management. A well thought out plan should be made, and tight discipline enforced. A key management scheme which is loosely handled will produce chaos and could result in a reduction in overall security. A trade-off should therefore be made between additional complexity and the need for smooth day to day operation.

3. Key Generation

Keys themselves should be unpredictable and changed as frequently as necessary (based on risk assessment). It may be better to change them at unpredictable times. It makes the criminal's job easier if he knows keys are changed at the same time on the same day each week.

Any temptation to relate keys to other entities (such as names, dates, I.D. numbers) should be avoided. Neither should keys be chosen so as to form an easily memorized sequence of characters. This would limit the number of usable keys to a quantity far less than the maximum. The keys should be generated so as to be statistically independent and uniformly distributed over the range 0 to 2^{56} i.e., there should be an equal probability of any key being generated as shown in figure 1. Computer programs which always generate the same sequence of random numbers obviously should not be used. Instead, programs using a variable seed obtained from an external source provide a much superior method. Note that the key generation must be done on a 56 bit basis and 8 parity bits added subsequently because a 64 bit random number would be rejected by many encryption devices if the key parity check failed. Figure 2 shows a simple scheme for generating keys. A

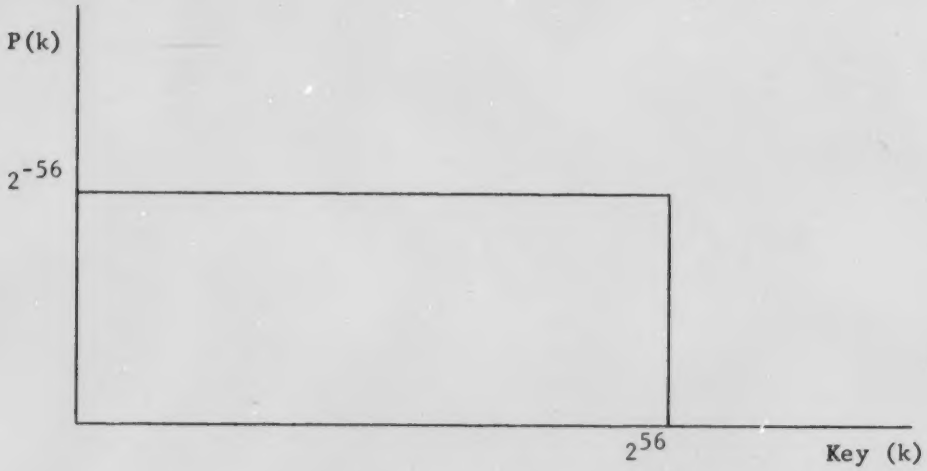


Figure 1 Uniform Distribution of Keys

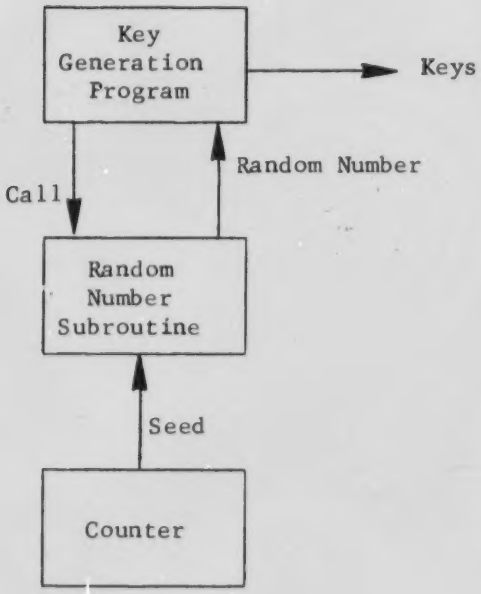


Figure 2 Key Generation

hardware counter running at several hundred kHz and in no way synchronized to the processor is read by a random number subroutine. The counter contents at the time of reading are used as the seed for the random number generation. Since the time at which the subroutine is called is random relative to the counter, the seed is totally unpredictable. In many systems the time of day clock can be used as the source of a seed.

Random number generation is a subject in itself. Reference [1] gives a good overview of the topic and also discusses methods of testing the randomness. This reference also contains an extensive set of further references.

Needless to say, the key generation program must itself be carefully scrutinized to ensure there are no inputs or outputs other than the intended ones. Also, the key generation program must be run under strict supervision and memory used during the key generation process should be erased after use.

4. Key Storage

Once keys have been generated they should be stored in a protected area of memory until use. They should not be printed out unless absolutely necessary. The time keys are in storage should be minimized by generating them as late as possible. If long term storage is unavoidable (as in the case of file encryption applications) the keys themselves should be encrypted with another "master" key. This latter key should not be resident in any part of the system.

5. Key Loading

There are four basic methods by which a key can be loaded into the encryption device. Not all are available in the marketplace; they only indicate possibilities.

5.1 Manual Switches

Most of the first available products will use this approach. 16 hexadecimal switches can determine 64 bits. Since this 16 hex digit number will be difficult to remember, it must be written down on paper which must be properly safeguarded. The devices should be locked up out of sight within a secure area.

5.2 Plug-in Modules

A small module containing read only memory can be used to convey the key to the encryption device. Once the ROM's have been programmed under strict security controls, the module can be handled and the key loaded into the device without anybody knowing the actual key. Furthermore, compared to a device with manual setting, changing the key is made much more difficult for the criminal.

5.3 Magnetic Stripe

This method is less expensive than the ROM above and still has the advantage that the key is not visible to the person handling it. The magnetic stripe reader may be built into a terminal device or can be in the form of a separate portable device only accessible to persons authorized to handle the key.

5.4 Electrical Interface

If the device is physically adjacent to (or built into) a communications processor, the key can be loaded via an electrical connection to the processor I/O. This enables the keys to be transferred from tables in memory to the device without human handling.

An extension to this method is the transmission of the key down a communication line to a remote encryption device. Obviously special precautions have to be taken in this mode.

6. Key Distribution

There are only three basic ways of distributing keys:

6.1 Registered mail with its attendant risks.

6.2 Courier, which for a price, can be as secure as desired.

6.3 Down line load which is very dangerous unless the new key is encrypted with a special key which is never transmitted over the line. Encryption of the new key solely by the current key is not recommended for obvious reasons.

One way <u>not</u> to transmit keys is verbally over the telephone. One may become so preoccupied with the security of the data link that a little carelessness when talking on the telephone could easily give away the key.

7. Link Encryption

Link encryption is probably the method most users will elect to use in their first encryption applications. This is because it will be the method which has the minimum impact on hardware and software in existing systems. Keys will be set manually in most cases, and the rules mentioned earlier must be observed.

If dedicated lines are used, which is the preferred way, there should be a different key for each link, and possibly a different key for each direction of traffic on the same link.

If dial-up lines are necessary because of a high ratio of terminals to ports on the communications processor, then each terminal should have its own key. Figure 3 illustrates a subset of such a system.

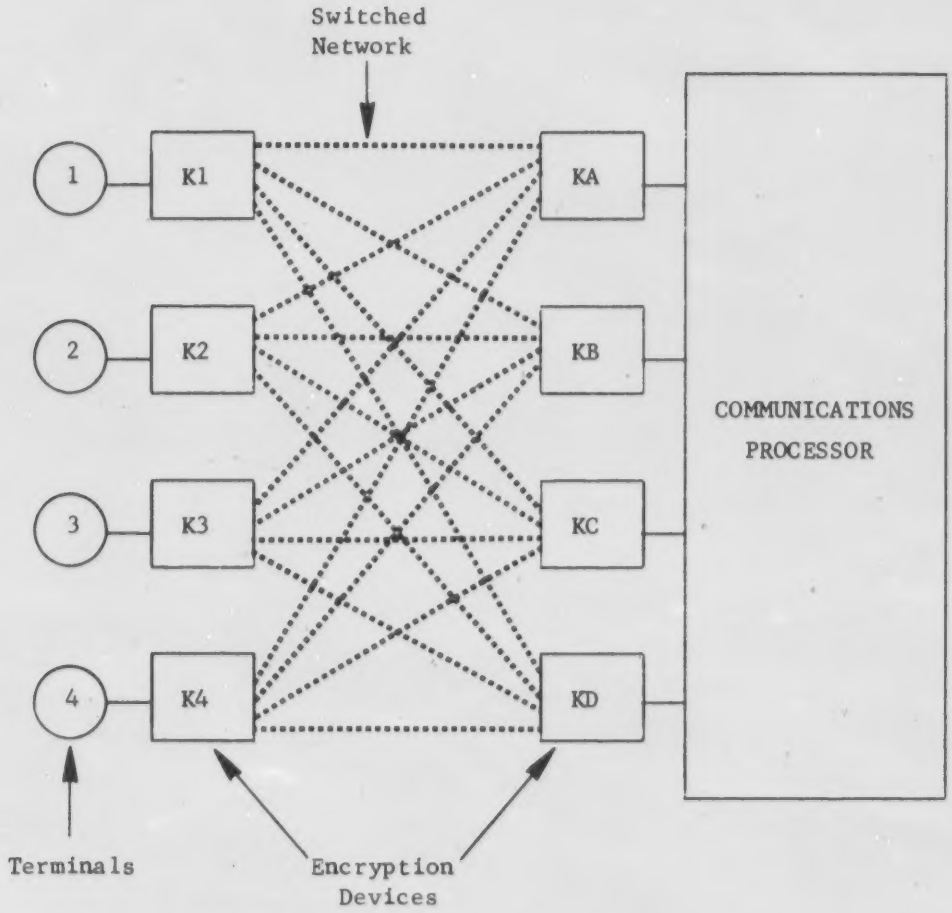


Figure 3 Key Management in a Switched Network

The keys K1 through K4 have been previously inserted in the terminal encryption devices, and a table mapping I.D. and keys is stored at the central site. A terminal first identifies itself in the clear, and this enables the network processor to set the appropriate key into the device associated with the port to which the dial-up connection is made. When the call is completed, the key is erased from the device at the central site.

8. User Oriented Keys

In some cases it may be beneficial to have user oriented keys, instead of, or in addition to, device or link oriented keys. With this approach, each user has his own key. He may or may not know the actual key depending on the form of handling.

In a EFTS application the key can be in the form of a Personal I.D. number (PIN) which is used to encrypt the Personal Account Number (PAN). The resulting encrypted PAN is then further encrypted by using a device oriented key. If the PIN has to be entered via a keyboard, the user must know the PIN.

Another scenario for user oriented keys is where a high level of security is required. Each user has a key on a magnetic card, and the card is surrendered to the guard as the user leaves the secure area where the terminals are located. The user does not know the key nor does he know when it has been changed. To gain access to the system, he first identifies himself in the clear and then after insertion of his key, switches to encrypted mode. He then enters his own password, date and time of day which are encrypted and transmitted to the central site. He is only permitted to continue his dialog if the password decrypted by the key assigned to him checks with the password on file. The time and date is also checked to guard against the possibility of a recorded message being played back into the system at a later time via an active wiretap.

9. Composite Keys

In cases where very special precautions have to be taken, the concept of composite keys can be employed. The actual key used is derived from one or more keys by some simple process such as modulo 2 addition. The encryption equipment must be designed so as to perform this operation prior to loading of the actual key into the encryption chip. The individual keys can be handled by separate persons or one key can be user oriented and the other device oriented. Each key must be the full length. Giving half the key to one person and half to another would drastically reduce the security level since the ratio 2^{56} to 2^{28} is the same as the ratio of 1000 years to 10 minutes.

10. Summary

Key management schemes must be tailored to the needs of the individual organization. One can conceive of "ultimate" solutions using end to end encryption with key generation and loading performed automatically by a computer assigned to the task. It will be several years before such schemes can be considered a reality, and in the meantime we will have to use more down to earth approaches. Human beings will be heavily involved in key management, and as in any security situation, careful steps must be taken to ensure their integrity.

In practice it will be necessary to sacrifice extra complexity for the sake of smooth operation. In addition to careful planning, the chosen system should be thoroughly tested and particular attention paid to what would happen in abnormal situations such as loss of a key or recovery from a system crash.

All possible eventualities should be considered and a comprehensive set of rules established. A tight discipline must then be enforced.

A final reminder, if the key management scheme is not designed properly or adequately enforced, the result could not only be disastrous from a security viewpoint but the viability of the entire system may be jeopardized.

Reference

[1] Chambers R.P., Random Number Generation, I.E.E.E. Spectrum, February 1967.

Design and Specification of Cryptographic Capabilities

Interbank Card Association
Carl M. Campbell, Jr. (Consultant)
809 Malin Road, Newtown Square, Pa. 19073

Cryptography can be used to provide data secrecy, data authentication, and originator authentication. Non-reversible transformation techniques provide only the last. Cryptographic check digits provide both data and originator authentication, but no secrecy. Data secrecy, with or without data authentication, is provided by block encryption or data stream encryption techniques. Total systems security may be provided on a link-by-link, node-by-node, or end-to-end basis, depending upon the nature of the application.

Key words: Cryptography; data security; encryption.

1. Introduction

Up to the present, cryptography has been a relatively unknown science, used primarily to secure sensitive governmental communications. However with the introduction of the Data Encryption Standard (DES) we expect to see cryptography widely applied in data processing systems, especially in digital communications, to provide data security. It is thus essential that the designers of these systems gain an understanding of this new technology.

2. Uses of Cryptography

Cryptography can be used to provide three aspects of data security:

(1) Data secrecy.

(2) Data authentication.

(3) Originator authentication.

The first use of cryptography, data secrecy, is relatively well understood, and will be an important use in an EDP environment.

Data authentication and originator authentication are less understood, but will be very important uses of cryptography in the future. To understand data authentication, assume that "A" is transmitting data to "B." "B" wants assurance that the data it is receiving is precisely the data which "A" transmitted. Though conventional error control techniques can protect against communications errors, "B" is concerned that someone with a sophisticated "active wiretapping" capability may have deliberately modified the data from "A," and made the appropriate modifications in any associated error control fields. Cryptographically-implemented data authentication provides assurance that the data was received as originated.

Originator authentication is similar to data authentication. This time "B" requires assurance that it is receiving data from the "real 'A'" and not from an impostor who may have assumed "A's" identity. Again, cryptography can provide the solution.

There are an almost unlimited number of ways in which cryptography can be applied. Some applications meet only one or two of the above objectives, and some meet them all.

3. Originator Authentication

A simple use of cryptography meets only the third objective, originator authentication. In this approach, figure 1, each authorized user of a system is given a secret "authorization code." Each terminal incorporates a cryptographic capability into which he enters this code. The code is "non-reversibly transformed" into another code. This means that, given the transformed code, there is no way to determine the actual code except for an exhaustive "trial and error" procedure, which is presumed to be non-feasible if the original code is quite long (approximately 56 bits) and reasonably random. The system's central processor stores, in a manner which may be non-secure, each user's transformed code. A simple comparison is thus sufficient to authenticate the user.

Note that this approach does not require a unique terminal key, so imposes no "key management" requirements. Note also that it does not require any on-line cryptographic capability at the central facility.

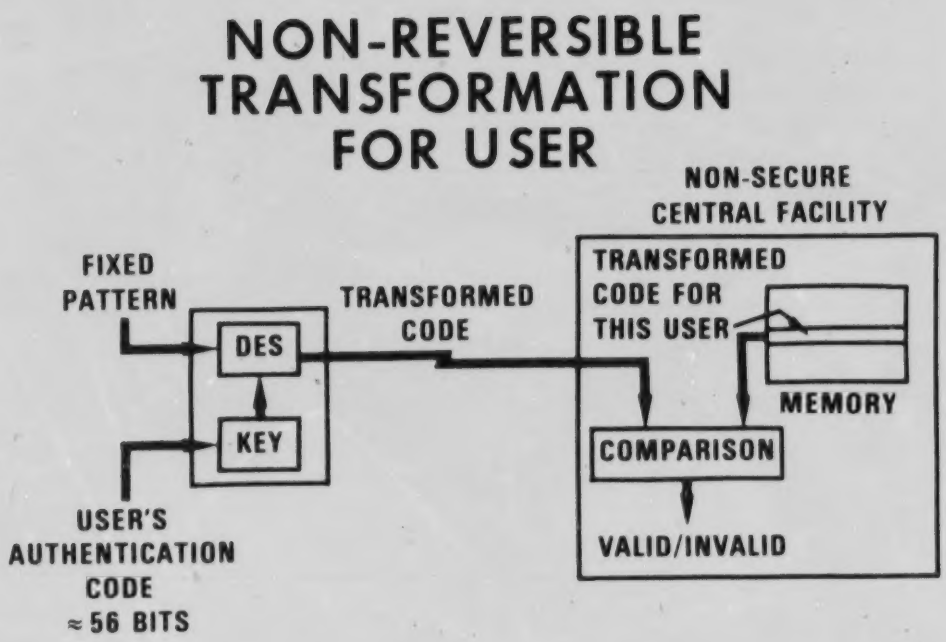

NON-REVERSIBLE
TRANSFORMATION
FOR USER
NON-SECURE
CENTRAL FACILITY
FIXED
PATTERN
DES
KEY
TRANSFORMED
CODE
TRANSFORMED
CODE FOR
THIS USER
MEMORY
COMPARISON
VALID/INVALID
USER'S
AUTHENTICATION
CODE
≈ 56 BITS

Figure 1.

4. Data Authentication

A very useful cryptographic technique, cryptographic check digits, provides data authentication and can provide originator authentication, but provides no data secrecy. Cryptographic check digits may be likened to parity check digits or to a cyclic redundancy check in that a check field is added to the message by the originator and verified by the recipient. However, unlike a conventional error-control check field, the cryptographic check digit field is generated by a cryptographic algorithm and utilizes a secret key known (desirably) by originator and recipient alone. Thus the field protects not only against accidental garbles, but also against deliberate attempts to modify the transmitted data. Without knowing the secret key, the one attempting such data modification would be unable to make the appropriate changes in the cryptographic check digits field which would be required for his modification to escape detection.

Note that originator authentication is provided if the recipient is certain that only the authorized originator posesses the secret key.

DES may be used to generate cryptographic check digits, as, for example, is illustrated in figure 2. Each group of 64 message bits is passed through the algorithm after being combined with the output of the previous pass. The final DES output is thus a residue which is a cryptographic function of the entire message. All or part of this residue may be used as the cryptographic check digits.

Cryptographic check digits alone cannot detect the fraudulent replay of a previously valid message, nor the deletion of a message. To protect against these threats, each transmission of a message must be made unique. One technique is to insert a cryptographically-protected sequence number into the message. Another is to use a different key for each message.

5. Data Secrecy

Secrecy of transmitted data may be provided by a number of techniques, some providing data authentication and some not. All of the suggested techniques utilize a secret key, and so provide originator authentication if this key is properly controlled.

5.1 Block Encryption

The Data Encryption Standard is inherently a block encryption algorithm, requiring blocks of precisely 64 bits.

CCD GENERATION

DATA (ORIGINATOR) AUTHENTICATION

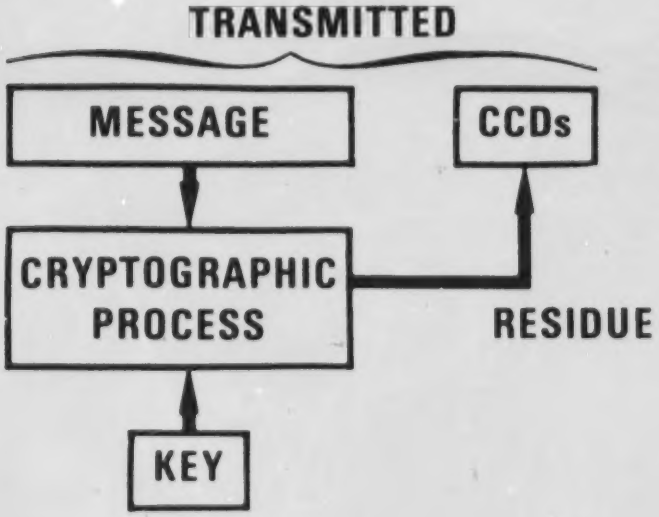


Figure 2.

Given a plain-text block of 64 bits, a secret key, and the "encrypt" command, the DES algorithm produces 64 cipher bits. Given these 64 cipher bits, the same key, and the "decrypt" command, the algorithm produces the original 64 plain-text bits. Thus, as long as the block size is exactly 64 bits, block encryption with DES is extremely simple.

Short blocks. If the block size is less than 64 bits, these bits must be "padded" (with any fixed or variable pattern) to make 64 bits if the algorithm is to be used in its normal block-encryption manner. All 64 of the resulting cipher bits must be transmitted to the recipient even though only 20 bits of underlying information are present. The recipient block-decrypts these 64 bits, resulting in 64 plain-text bits. All but 20 of these must be discarded, leaving the 20 original information bits.

The use of DES for a block size of less than 64 bits is thus somewhat inefficient, in that the full 64 bits must still be transmitted. Different techniques for using DES are possible, which overcome this disadvantage, but they introduce other disadvantages.

Multi-blocks. Where the block to be encrypted is long, it can be broken up into groups of 64 bit blocks, and each such block encrypted independently. This simple approach provides secrecy, but it does not provide a high degree of data authentication. For example, assume two block-encrypted messages, one reading: "PAY TO J. JONES $9,000.00" and the second: "PAY TO S. SMITH $1,000.00." If the "$9,000.00" and the "$1,000.00" should each fall precisely within a block, it would be possible to replace the cipher block for "$1,000.00" with that for "$9,000.00" so that when the recipient decrypts the second message it reads: "PAY TO S. SMITH $9,000.00."

This process, by which cipher is manipulated, is called "spoofing." Note that the "spoofer" knows corresponding cipher and plain text, but does not know the secret key. His objective is to intercept, modify and then retransmit the cipher, all in such a manner that his deception is not detected.

Encryption techniques can be devised which prevent "spoofing," but in order to do so it is necessary to introduce something called "garble extension." This means that if any portion of the cipher becomes garbled (i.e. changed) the decryption by the recipient of a certain amount of subsequent cipher is also garbled.

Figure 3 illustrates one method by which garble extension, and hence spoofing prevention, can be incorporated into a block encryption system. The "E" boxes perform block encryption, and the "D" boxes block decryption. The "+" function indicates exclusive-or. The approach of figure 3 provides "infinite" garble extension. That is, any change to the cipher garbles the decryption of all subsequent cipher. Infinite garble extension has the features that the originator can place in the final block a pattern expected by the recipient. If the recipient finds the expected pattern at the end of the message, he is assured that the entire message, regardless of length, was received precisely as originated.

5.2 Data-Stream Encryption

The term "data-stream" refers to the serial flow (serial by bit, by character, or any other increment) of data, as over a communications line. "Data-stream encryption" refers to the encryption of such data in real-time, for subsequent "data-stream decryption," also in real-time. It is possible to use block encryption for data-stream encryption, but this is not desirable. In DES block encryption, the first bit cannot be encrypted until the 64th bit has been received, so that a block-encryption technique in a data-stream environment inherently imposes a delay of 64 bit times. Block decryption imposes an equal delay. Thus, communications delays would be unacceptably increased where block techniques are to be used.

Fortunately, DES can be applied to a data-stream environment so as to minimally impact communications delays. Two such techniques are "internal feedback" and "cipher feedback."

Internal Feedback. The internal-feedback approach to data-stream encryption uses DES to generate a stream of pseudo-random "encrypting bits." These bits are exclusive-ored with the plain-text bits to form the cipher bits, as illustrated in figure 4. The decryption process operates the same way, with the exact same pseudo-random stream of "encrypting bits" being generated. Exclusive-oring these bits with the cipher bits then produces the original plain text bits.

To use DES in this manner, any number of the 64 output (i.e. cipher) bits may be used. For simplicity of explanation, it is assumed that only 1 bit is used, and the other 63 discarded. The selected bit is not only used to encrypt the plain-text data, but is also fed back as the input to DES, and another algorithm cycle initiated. Thus, one algorithm cycle is required per "encrypting bit."

BLOCK INTERCONNECTIONS
TO PROVIDE "INFINITE " GARBLE EXTENSION

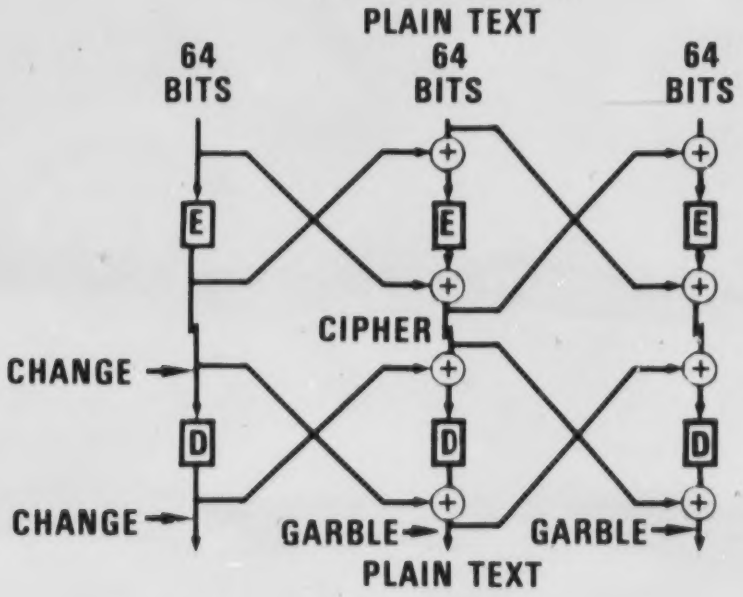


Figure 3.

INTERNAL FEEDBACK

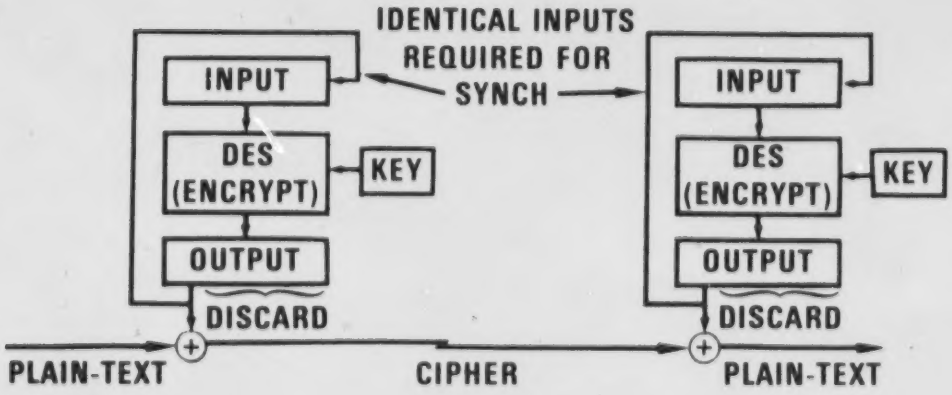


Figure 4.

To ensure that the decryption process generates the same pseudo-random "encrypting bits" as does the encryption process, the DES input registers of the two devices must commence operation with the same "initial fill." The process by which this is accomplished is called "crypto synchronization."

Cipher feedback. This approach to data-stream encryption is very similar to the internal feedback approach, the difference being that cipher bits, rather than "encrypting bits," are used as the DES input. Note that this approach, Figure 5, if used in a one bit feedback mode, is "self synchronizing" because after 64 bit times the DES input register of the decryption device will contain the same data as does the input register of the encryption device. Note also that the approach provides garble extension, thus providing anti-spoofing protection.

6. System Philosophies

There are three basic approaches to incorporating encryption into a communications system: link-by-link, node-by-node, and end-to-end encryption.

Link-by-link encryption, figure 6, is the technique most commonly used today. It may be implemented in a transparent manner with currently available devices, which are placed in series with the circuit between data terminal equipment and data communications equipment. This approach has the disadvantage that it allows all traffic to pass through the CPU of any node in plain-text.

Node-by-node encryption, figure 7, is a modified version of link-by-link encryption to overcome this disadvantage. Each link uses a unique key, but the "translation" from one key to the next occurs within a single "security module" which might serve as a peripheral device to the node's CPU In this way plain-text data does not traverse the node, but exists only within this physically secure module. Note that enough message data must remain encrypted so that the node's CPU can properly route the message.

End-to-end encryption, figure 8, requires a "Key Control Center," located somewhere within the communication system. Each end-point in the system holds a unique "long-term" key, and this center alone holds a copy of each such key. When one end point wishes to communicate to another, a request to this effect is sent to the Key Control Center. This center then generates a temporary "per conversation" key, encrypts this in the long-term key of originator and also in long-term

CIPHER FEEDBACK

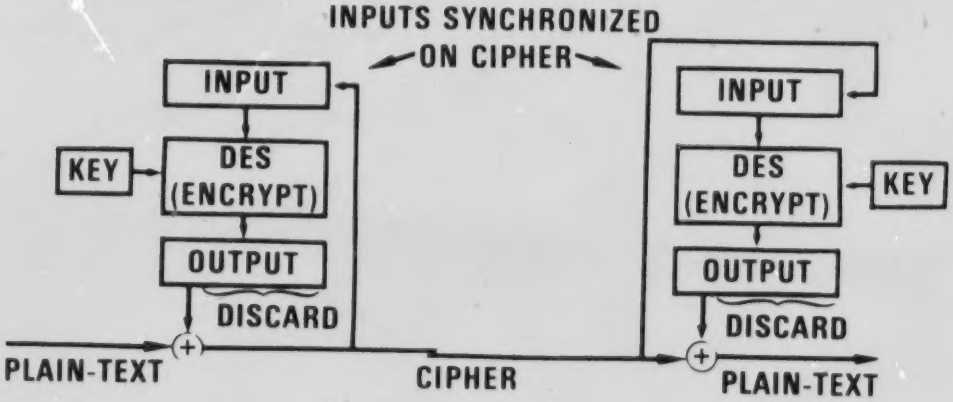


Figure 5.

LINK-BY-LINK ENCRYPTION

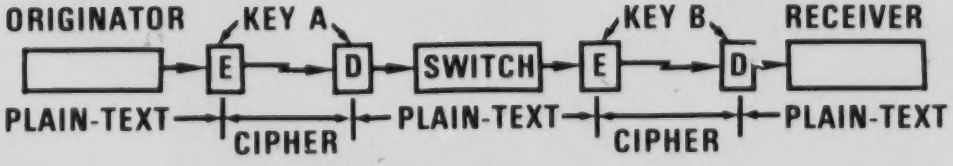


Figure 6

NODE-BY-NODE ENCRYPTION

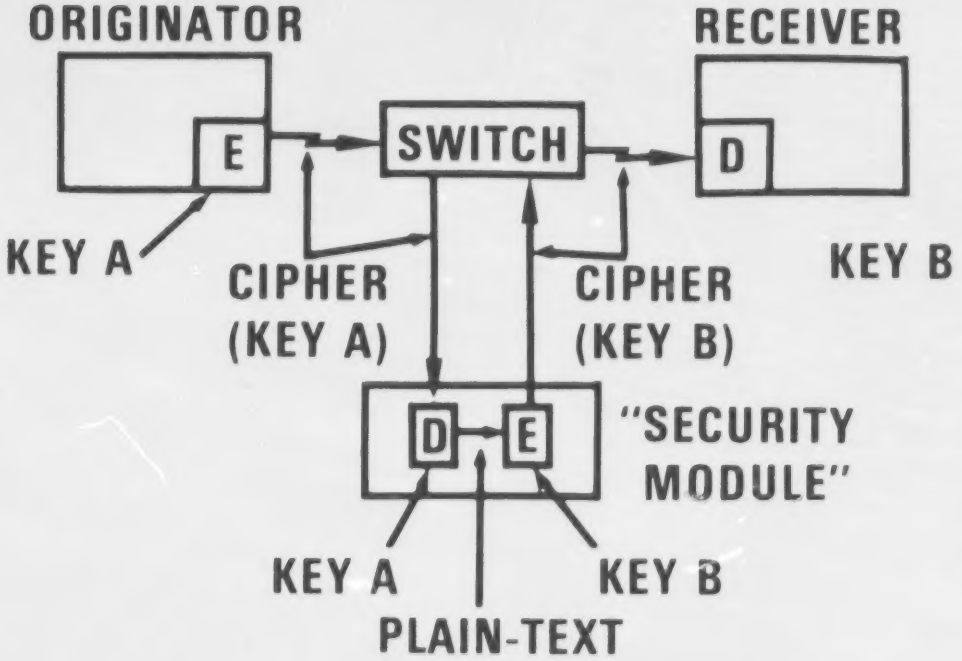


Figure 7.

END-TO-END ENCRYPTION: CONNECTION SET-UP

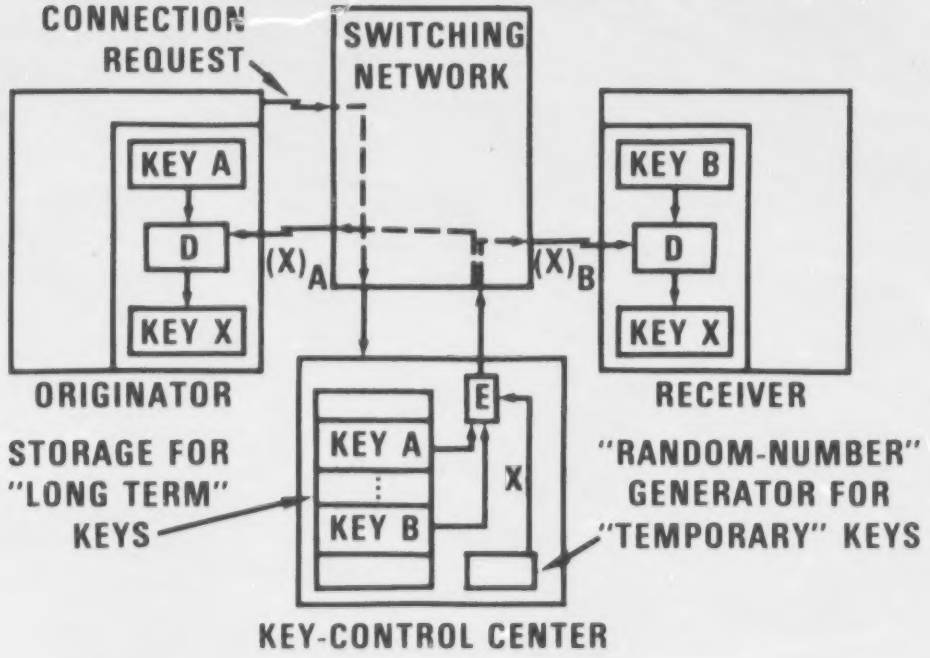


Figure 8.

key of the recipient, and sends the appropriate version to each. The originator decrypts this just-received encrypted temporary key using its long-term key, the recipient does likewise with its long-term key, and the two parties then converse with end-to-end encryption using this temporary key.

7. Procurement Considerations

For retrofitting an existing system, link-by-link encryption utilizing transparent link encryption devices is a reasonable approach. DES feedback is a desirable choice for these devices.

For a new system, in which cryptography can be "designed in" rather than "added on," block-encryption techniques should be considered because of their more efficient use of the algorithm, and their absence of initial synchronization requirements. For a transaction oriented system, in which messages are very short and routed to varying destinations, the node-by-node approach appears preferable because it does not impose any per-conversation overhead for key distribution. However for a "session" oriented environment in which conversations may be relatively long, end-to-end encryption appears to be the obvious choice.

References:

1. Branstad, Dennis K., "Encryption Protection in Computer Data Communications Systems," Fourth Data Communications Symposium, Quebec, Canada, October 7-9, 1975.

2. Kent, Stephen T., "Encryption-Based Protection Protocols for Interactive User - Computer Communications," Technical Report 162, Laboratory for Computer Science, Massachusetts Institute of Technology, May, 1976.

3. Sykes, David J., "Protecting Data by Encryption," Datamation Magazine, August, 1976.

A Bit-Slice, 4-Chip Implementation of the Data Encryption Standard

Kris Rallapalli
Fairchild Semi-Conductor
Mountain View, California

The following paper has been extracted from the verbal presentation of Mr. Rallapalli at the February 15th Conference. A written paper had not been submitted at the time of publication of these proceedings.

1. Introduction

I would like to present an approach for implementing the DES in a bit-slice, multi-device, large scale integrated technology. This approach is based on our estimate of the user's need for a high-speed implementation of the DES for secure data communications. We feel that a high-speed hardware implementation can be widely used in many ADP security applications. The existence of a standard in this area potentially allows us to reach this goal.

2. Bit-Slice Implementation

We have attempted to design a set of chips which can be used in high-speed, cost effective applications in various environments having a wide range of temperatures. For this we have chosen to use the I^3L (Isoplanar Integrated-Injection Logic) technology.

It was quite easy to draw a block diagram of the DES. NBS did all of the work for us. After analyzing the requirements of the DES in a single chip, we felt that the chip would be far too large and expensive. In large scale integrated technology, the smaller the chip, the higher the yield, and hence the cheaper the cost. By analyzing the algorithm, we discovered that we could partition it into four parts. Each part could be implemented in one chip and all four chips would be almost identical.

After analyzing both the initial and final permutations of the DES, it became obvious that it would be simple to partition the DES in this way. The 64 bits of data are entered in eight 8-bit bytes. For each byte of data, device 1 would receive bits 1 and 2, device 2 would receive bits 3 and 4, device 3 would receive bits 5 and 6, and finally device 4 would receive bits 7 and 8. Eight bytes would be presented to the four devices in this manner until all 64 bits have been entered.

In the block diagram of the DES, the next major operation is to expand the right hand 32 bits to 48 bits. The next major operation is the XOR function of 48 bits of the key with the expanded right hand half. Each of the four devices will contain two substitution (S) tables. Device 1 will contain tables 1 and 2, device 2 will contain tables 3 and 4, etc. The four devices must be connected in such a way that they receive the necessary bits from the neighboring devices at the proper time to make the algorithm work. As far as the key is concerned, I am going to divide the key into 4-bit slices similar to the 2-bit slices used for the data. In order to do this efficiently in the four chip approach, I must maintain duplicate copies of the key across the four devices. In analyzing the DES, especially in the permutation of the key (PC-1 and PC-2), it is obvious that the C register must be in devices 1 and 2 and the D register must be in devices 3 and 4. The trick will be to input the key in 4-bit slices and to keep two copies. To control the devices, I propose two control lines. I am planning to use a microprocessor to control the four devices via the two control lines.

3. The 4-Chip DES

In summary, we are going to use four of these devices, where each device consists of two 8-bit shift registers for the data, four 8-bit shift registers for the key and two 64 X 4 ROM's for the S tables. Each device will have a parity check facility for the key and other required control logic. The four devices will work in parallel from a single clock. Our estimate of the speed is that it could be clocked at 5 megahertz. The two control lines that I mentioned would implement four control functions. The first is load key, the second is load data, the third instruction is to encrypt and the fourth is to decrypt data.

The device will check parity of the key as it is entered and set a flag for the microprocessor control if the parity is incorrect. It will not, however, prevent operating with a "bad" key. It takes eight clock pulses to load the key and eight more clock pulses to load the data. Then the devices require sixteen more clock pulses to either encrypt or decrypt the data, and eight additional clock cycles to unload the devices. However, the next eight bytes can be loaded at the same time that the unloading is taking place. Therefore, only twenty-four cycles are used for a complete operation of the DES unit. With a clock operating at 5 megahertz, this gives an effective throughput of 13 million bits per second, or in other words, each 64-bit block requires 5 micro-seconds to encrypt or decrypt.

Our company is planning to build these LSI devices and market them in various forms to our customers.

Federal Reserve Communications Security Project

Howard Crumb
Federal Reserve Bank
33 Liberty Street
New York, N. Y. 10045

The following paper has been extracted from the verbal presentation of Mr. Crumb at the February 15th Conference. A written paper had not been submitted at the time of publication of these proceedings.

1. Introduction

This afternoon I plan to discuss the Federal Reserve Communications System, some of our concerns for security, and the type of operations that the communications system supports. The Federal Reserve System was created by an act of Congress in 1913. Its job was to insure an orderly economic growth, supervise and regulate banks, act as a fiscal agent for the United States Treasury, and provide for an improved collection system. The United States is divided into twelve Federal Reserve regions and there is a Federal Reserve bank in each of the regions. Each bank is an independent corporation. The overall guidance for the Federal Reserve system comes from its Board of Governors located in Washington, D. C. However, each of the banks is responsible for its own operation.

2. FEDWIRE Communications System

With this introduction, I would like to talk about the communications system between these banks frequently referred to as the FEDWIRE. This system is used to transfer balances between Federal Reserve member banks throughout the country. There was a manual system before FEDWIRE was installed consisting of couriers which transferred money among the member banks, and as a result was vulnerable to those hazards and threats affecting physical transportation. The FEDWIRE was developed to eliminate charges for transfer of funds imposed by the courier system and to make the transfer of funds much faster.

The FEDWIRE consists of a central communications site at Culpeper, Virginia and communication lines to each of the Federal Reserve Banks. Similarly, each Federal Reserve Bank is linked to its member banks within its own region or district. FEDWIRE became operational in late 1970. At that time each Federal Reserve bank was

connected to Culpeper by teletype circuits. Subsequently, magnetic tape transfer capabilities were added to the twelve main communication lines.

The system was next upgraded by replacing the teletype circuits with computer communications switches. Each district was allowed to design, select and implement its own computer system but was required to meet standard interface criteria. Some of these standards in turn have been adopted for use within each district for interconnection to member banks.

Currently, the FEDWIRE system averages over 50,000 transfers per day, carrying well in excess of one hundred billion dollars. This is equivalent to transferring the Gross National Product every 15-18 days or transferring the National Budget every five days.

The Federal Reserve System has been fulfilling its role as fiscal agent by transferring Government securities for some time. The operation has evolved as a natural extension of FEDWIRE services to transfer the securities on a timely basis. This system has helped to eliminate much manual handling of Federal paper securities and in making this system much more efficient. Presently, about 83% of the National Debt is contained in this "Book Entry" form.

As a part of its fiscal responsibility, FEDWIRE is being used to transfer payrolls to approximately 250,000 Air Force personnel. These paychecks are being forwarded directly to many financial institutions across the country in a paperless form. This Air Force payroll is only a forerunner of a much larger operation. Concurrently, over five million Social Security payments are being transferred to Social Security recipients in a paperless form across the country. Other Government payrolls are planned to be converted to a paperless form in 1977.

In order to assure that network facilities will be able to handle these increased demands, we are planning to extend the system to handle this expanded load on a specified priority basis. In addition to expanding this system due to the increased load, we are planning to improve the security of the FEDWIRE. The FEDWIRE system must be protected for both availability and security reasons. The system must be available to make all the necessary daily transactions and these transactions must be protected against several threats and vulnerabilities. These vulnerabilities include sabotage, fraud and mischief. At present, significant controls exist to minimize these vulnerabilities. The security of the capability consists of physical security, operational security, personnel selection and network concerns, as well as the management aspects such as legal agreements and audit procedures.

We recognize that it is impossible to prevent all possible security problems. However, the system is designed to bring any exception to light as soon as possible after it occurs. We continuously monitor the operations to detect any fraud, accident or misuse. Our security

program is based on a cost effective approach in assuring users of high reliability and security to support daily operations.

3. Federal Reserve Communications Security Project

To assure the security of the Federal Reserve Communications System, we have undertaken a project to evaluate and improve the security of the network. It was initiated by publishing a request for information relating to security requirements of an electronic communications network. This request was issued in January, 1975. The principal purpose of the request for information was to seek those methods which would provide cost effective security for preventing modification, deletion or insertion of messages during transmission. The request for information was widely distributed throughout the country. The responses ranged from requests for support in developing software security, to well conceived approaches to network security. We were fortunate that the NBS algorithm was published for comment in 1975. Most of the responses to our request for information were encryption dependent and several recommended the use of the NBS encryption algorithm. Subsequently, several of those responding were asked if they would participate in a test of using the NBS encryption algorithm in hardware devices on the FEDWIRE network. The FEDWIRE Security Task Force decided that functional specifications had to be made by the group for all vendors as each vendor had different criteria that they felt should be met. The Security Task Force was not only interested in the security aspects of the devices, but also in their interfaces and their impact on the existing FEDWIRE system.

One of the major requirements for the security of the FEDWIRE system is to protect the encryption key used with the NBS algorithm. This protection must be provided during key generation, distribution, storage and in the operational device. Detailed guidelines for managing the encryption keys are in the development process.

The functional technical specifications required that the device be transparent to the existing communications system. The Task Force also required that using the devices would not imply any change to existing hardware or software in the network. The system consists of full duplex, half duplex, synchronous and asynchronous communications protocols. We desired that the device provide unattended service, prompt notification of abnormal operation, have no apparent impact on throughput, restart automatically, be easily installed and removed and be available in excess of 99% of the time.

The need was recognized very early that the device should be compatible with a variety of networks so that it can be easily retrofitted when others share our recognized need for transmission security. Testing of the prototype device is to begin in mid 1977. Upon receipt of these devices, there will be extensive testing to assure that the network and the security devices meet the stated requirements.

Evaluation criteria for the test and managing the encryption keys are currently being developed for the operational tests.

No specific action following the tests has been specified but it is hoped that commercially available devices will be offered to the Federal Reserve System and anyone else based on the results of this prototype system. We feel that encryption will also be needed in the future as one technique to meet requirements for privacy of information.

ARPA NETWORK SECURITY PROJECT

Stephen T. Walker
Defense Advanced Research Projects Agency
1400 Wilson Blvd.
Arlington, VA 22209

The ARPA computer network has become an operational Defense Department packet switched communications system. A recent ARPA research project has developed techniques for achieving end-to-end encryption processes in a sophisticated networking environment such as the ARPA network. The National Bureau of Standards' (NBS) Data Encryption Standard (DES) Algorithm has been employed as the basic encryption mechanism for the initial demonstration of this capability. This paper gives the background and current status of that research project.

A research project in computer networks initiated in 1968, by the Defense Advanced Research Projects Agency, pioneered the development and demonstration of packet switched communications systems. Today the ARPANET is one of the largest and most sophisticated operational computer controlled communications systems in the world. The network depicted in figure 1 extends from Hawaii to Norway with approximately 60 nodes and 120 host computers connected by 50 kilobit dedicated communication circuits. The ARPA network is now an operational Defense Department facility under the management of the Defense Communications Agency (DCA). While growth in terms of number of nodes on the network has leveled off in recent years, traffic on the network has continued to double yearly. In late 1976 average daily traffic handled on the network exceeded ten million packets per day.

The technology employed in the ARPA network has provided the foundation for DCA's common user data network, Autodin II. This system will be the major data communication network for the Defense Department in the 1980's and 90's. The ARPA network has also served as the basis for a number of commercial and private networks and many foreign systems.

The ARPA network has evolved from a basic research project to a fundamental component in the development of a wide variety of advanced computer science techniques. It has for the most part been associated with unclassified research organizations throughout the U. S., and with the exception of a recent limited capability to transmit classified information, it remains primarily a non-secure facility. However, a major concern from the inception of the ARPANET has been the need within the Defense Department for efficient secure data communications mechanisms. Developing techniques for securing packet switched networks is the principal research objective of ARPA's network security program.

In the implementation of secure computer systems there are basically three levels of complexity to be considered: physical/administrative, communications and operating system (or software) security measures. Computers have been processing classified information for many years in what is called "system high mode" where the computer is physically isolated in a protected area and all personnel associated with the computer are cleared to the highest level of classified data processed by the system. The first level of complexity consists of the well known physical and personnel security measures, involving locks, alarms and clearances. When two or more secure computer systems are linked over communication lines, the second level of complexity, communications security, is employed. The universally accepted approach to communications security is the use of encryption on data while it is being transmitted over unsecured communication lines. Communications security techniques have been employed for many years in link encryption mode where each end of the communication line is attached to an encryption device. Both administative and communication security measures are used to protect computer systems from unauthorized external access.

The third level of complexity influencing the use of computers handling classified material is the operating system or software security problem. In this case the integrity of the software running in the computer must be relied upon to provide protection among authorized users of a computer system. A special case of operating system security is the control of encryption devices operating in a sophisticated networking environment. The computer controlled nature of advanced communications systems requires solutions to the security problem in addition to the already existing communications security issue.

The ARPA System and Network Security Program is addressing the third complexity factor described above. Considerable progress is being made in the operating system security area with the application of several certified secure ADP systems in the Defense Department anticipated within the next one to three years. A particular concern of the government, being addressed by this ARPA research program, is the employment of computer controlled encryption techniques to provide communications security within sophisticated computer networking environments.

In mid-1975 ARPA, in conjunction with other government agencies, began an effort to provide an effective demonstration of end-to-end encryption with remote key distribution. The basic concepts of this approach were first published in a paper by Dr. Dennis Branstad in 1973 (1). The system is designed to work in multiple networking environments allowing the encrypted data to pass unaltered among several interconnected networks. The system uses the newly developed transmission control protocol by Cerf and Kahn (2) to provide a highly reliable communications path. It makes heavy use of the layering effects of network protocols, insuring an essentially error free environment regardless of the communication path being employed.

For this demonstration system it was desirable to work with a sophisticated "real" encryption algorithm. With the announcement early in 1975 by NBS of the proposed DES algorithm, it was decided to employ this algorithm for the demonstration system in order to achieve a sophisticated demonstration in a minimal amount of time.

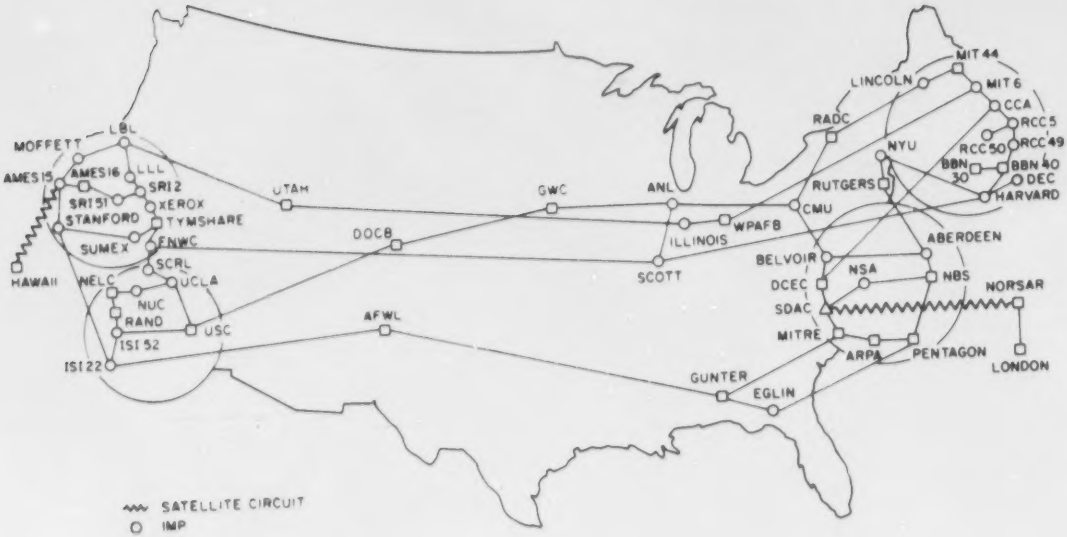
The basic hardware elements of the system are the encryption control units (called BCR boxes) consisting of two LSI 11 minicomputers controlling the basic encryption functions, and a key distribution center currently being developed on a PDP-11/40 minicomputer. Figure 2 illustrates the basic system which is presently being employed in a demonstration of the functions of the encryption system. In a typical scenario, the user activates his terminal and inserts his unique key variable card into the BCR box. The key distribution center is notified of the user's identity by the unique variable and dialog is begun with the user asking him to type a password. Once this initial authentication has been completed, the user specifies the destination he would like to reach on the network. The key distribution center checks that the user is authorized to access this facility and then initiates a separate dialog with that facility to insure that it is ready to accept the new connection. If all authentication checks are successful, then the key distribution center generates a unique key for this conversation, transmits it, encrypted, to both parties and authorizes them to establish communication using this new key. At any point the key distribution center can disallow the communication through its control of the key storage process within the BCR device. Once the key associated with this unique conversation has been distributed, the two ends of the conversation establish communications in the same manner that they would on unclassified networks such as the ARPANET.

Figure 3 illustrates a possible configuration for multiple BCR units employed on the ARPA network. Some form of a "secure" subnetwork of this type will be established in the Summer of 1977 for purposes of rigorous checkout of the protocols necessary to allow this end-to-end encryption process.

[1] Branstad, D. K., Security Aspects of Computer Networks, AIAA Paper #73-427, Computer Systems Conference-Huntsville, AL/Apr 16-18, 1973

[2] Cerf, Vinton & Kahn, Robert, A Protocol for Packet Network Intercommunication, IEEE Transactions on Communications Vol Com-22 #5 May 1974

ARPANET GEOGRAPHIC MAP, NOVEMBER 1976



(NOTE THIS MAP DOES NOT SHOW ARPA'S EXPERIMENTAL SATELLITE CONNECTIONS)

NAMES SHOWN ARE IMP NAMES, NOT (NECESSARILY) HOST NAMES

Figure 1

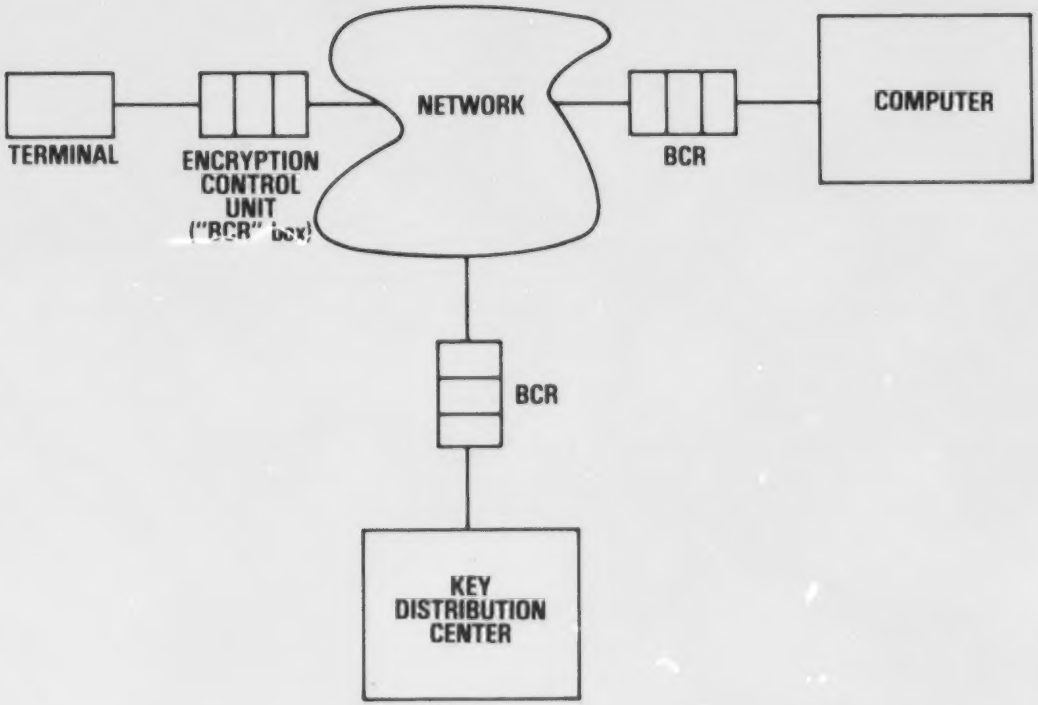

TERMINAL
ENCRYPTION
CONTROL
UNIT
("BCR" box)
NETWORK
BCR
COMPUTER
BCR
KEY
DISTRIBUTION
CENTER

Figure 2

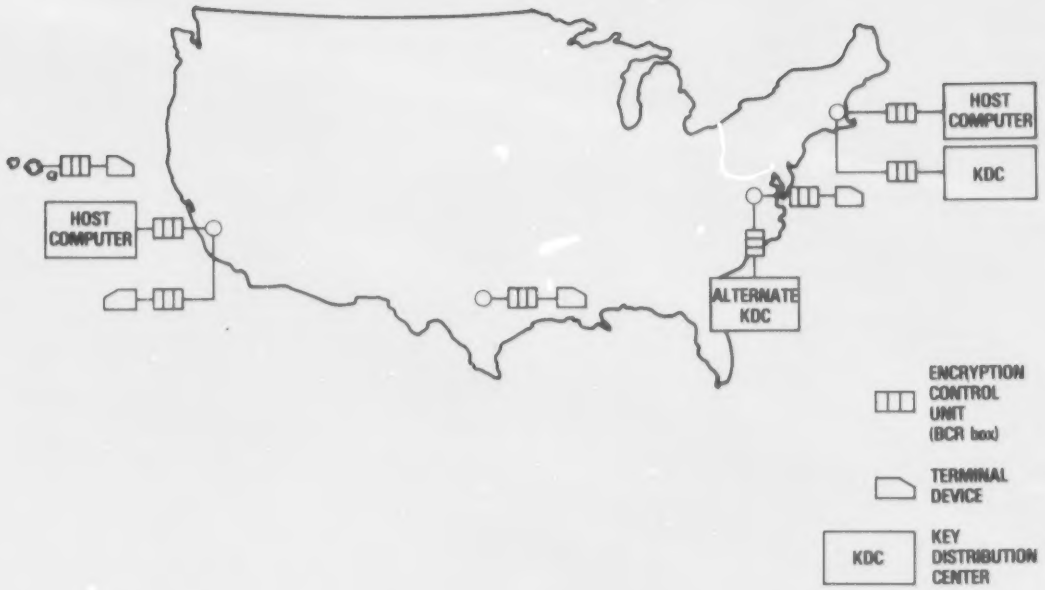

HOST COMPUTER
KDC
HOST COMPUTER
ALTERNATE KDC
ENCRYPTION CONTROL UNIT (BCR box)
TERMINAL DEVICE
KDC
KEY DISTRIBUTION CENTER

Figure 3

Electronic Funds Transfer Application

Jack McDonnell
EFT Commission
1000 Connecticut Avenue
Washington, D.C. 20036

The following paper has been extracted from the verbal presentation of Mr. McDonnell at the February 15th Conference. A written paper had not been submitted at the time of publication of these proceedings.

1. Introduction

I would like to preface my comments on security by introducing you to the National Commission on Electronic Funds Transfer. The EFT Commission was created by an act of Congress two years ago because Congress anticipated that there would be a lot of problems in the "checkless" society that do not exist in the present banking environment. Congress created this Commission to study these problems and report back with recommendations on what to do. Our first report is due to Congress on February 23, 1977. This will be an interim report and makes only non-technical recommendations. The final report will include our technical recommendations.

EFT is not new. The Federal Reserve has been using this mode of balancing the nation's "checkbook" for some time.

There are three main areas of EFT. The first I will call a low-volume, high-dollar transaction system typified by the FEDWIRE system. The second is the system typically called the automated clearing house, which primarily uses magnetic tape to transfer money. The third is the one I would like to discuss today; it is the one which has a high visibility to the consumer. The last incorporates automatic cash issuing terminals, point of sale terminals and automatic teller machines.

I would like to give credit for most of the material that I am going to present to Mr. Paul Havener of the Federal Deposit Insurance Corporation who has written a booklet entitled "Introduction to EFT Security." Figure 1-1 of this document displays the various points of vulnerability in an EFT system. In particular, the automatic teller machine is the direct interface of an EFT system to a customer. The customer must present a "digital signature" to the machine to prove the customer's identity. This digital signature is called a Personal

Identification Number (PIN). Typically, a customer is issued a plastic card with a magnetic stripe on the back in conjunction with the PIN. This magnetic stripe contains information in a 1, 2 and 3-track format. The combination of the card and the PIN causes the system to operate.

The effective use of encryption in the EFT environment requires several things. First, the banking community and its customers must be educated to the threats of an EFT system and the use of encryption in reducing these threats in order to establish a viable National EFT system. Second, the encryption of the PIN or other information on the plastic card requires several standards in order to be viable. Third, these standards must be available on a non-proprietary basis to be used at will throughout the system.

2. Threats to an EFT System

A cash issuing terminal usually has between twenty and forty thousand dollars at the beginning of a day. The largest "rip-off" that has been identified in an EFT environment did not occur in this country but in Switzerland. A customer with a valid card and a valid PIN used his knowledge of the off-line system to perpetrate his crime. He simply started visiting each of the cash issuing terminals in a large European City starting at 5 a.m. on a weekend to "jackpot" each of the terminals. To the best of our knowledge, he acquired the equivalent of $100,000.

I would like to look at the vulnerabilities of an EFT system and see where encryption can alleviate some of the potential risks. One application is to encrypt the data on the magnetic stripe of the card. If the PIN is used as part of the key for the encryption operation, anyone who finds or steals the card, but does not know the PIN, cannot use the card.

The simplest threat to the communications of an EFT system is passive wire tapping. In this threat a penetrator simply records the information going across the communication line and duplicates the magnetic card from the information contained in a transaction request to commit fraud. If the PIN or other input data of this communication were encrypted, the penetrator would be thwarted in this attempt.

The second threat is called active wire tapping. A penetrator is not only able to monitor the communications between a cash issuing terminal and a bank, but is also able to modify the communications.

We can look at encryption as being a security measure for communications. Unlike a simple communication system in which all of the data and central information is encrypted, a viable EFT network requires that only the valuable data be encrypted and the address/control information remain in the clear. This latter information is required in the switch between communicating devices. We feel that the Cryptographic Check

Digits (CCD)* hold great promise in securing an EFT network.

The final threat exists within the computer of each financial institution. I cannot emphasize too strongly that this is the most vulnerable point in any system. We feel that there is a definite application for encryption on the account files within the computer itself.

We hope to develop a set of security guidelines for the financial community through an inter-agency task group that we have established. It is too early to tell exactly how extensive these guidelines will be. Our first step is to inventory the cases of fraud that have occurred in EFT systems. Before the Commission is terminated in October, we hope that we can have a set of guidelines for financial institutions to enforce. In all probability, we will make the recommendation that this inter-agency task group continue in some form, perhaps in conjunction with NBS, to develop the technical security standards needed for an EFT system.

---

*Editor's Note: See the paper by Carl Campbell in these proceedings.

Implementation & Use of The Data Encryption Standard within The Data Communications Environment

Mr. Ed. Lohse
Corporate Engineering Headquarters
Burroughs Corporation
World Headquarters Building
Room 5E30
Burroughs Place
Detroit, Michigan 48232

With the standardization of the DES, product and system designers can proceed to implement various security devices. Applications for link and end-to-end protection can and will be accommodated. However, if these applications are likely to involve communication within a system containing equipment from different manufacturers, additional standards are needed: key management, electrical interface, encryption mode, initialization and resynchronization. This standards development effort is already started.

Key words: Encryption; security devices; standards.

With the advent of Electronic Funds Transfer systems, and data banks filled with statistics on individual citizens and businesses there is a growing inter-dependence between computer systems and communications systems. The transfer of this information to or from remote system users while maintaining the integrity of the data is in itself a complex problem. The passage of the Privacy Act of 1974 further compounded the problem by requiring that this information transfer cannot be accessed by unauthorized personnel. Beyond the need for privacy there is need to protect against alteration of the message.

This becomes doubly important when data is transmitted via common carriers such as microwave transmission systems, communication satellites or telephone lines.

It is encumbent upon the management of these user systems to guarantee the privacy of this information and guard against its fraudulent use or alteration.

The Data Encryption Standard (DES) has gone a long way in providing a tool with universal applicability for those who wish to ensure data protection.

The type of security required in some environments may be such that the message can be transmitted in clear text as long as its integrity is safeguarded. Other environments may require the contents of the message be concealed during transmission from unauthorized observation. In the former case authentication will suffice, that is, the message text is operated upon by the DES to produce a series of check digits which are appended to the message and this entire format transmitted. If, at its destination, the message integrity has been preserved the same set of check digits will be generated and a simple comparison will serve to validate the message. In the latter case, the text of the message will be transformed, using the DES, into cypher which is transmitted. This process is known as encryption.

These levels of security may be implemented in either of two data communications modes: link or end-to-end.

Figure I illustrates the levels of protection provided for each technique used.

Figure IIa and IIb show how these techniques are implemented in some data communication networks.

In the link mode the device is transparent to the data on the line, encrypting and decrypting without modifying any of the data in the process and without affecting the source or destination processors.

Data on the line between the devices is protected for both message integrity and secrecy (privacy) since it is unintelligible to unauthorized listeners and cannot be altered without detection.

In some communication networks with multiple nodes, link encryption does not protect the data within the node where the message is in plain text and subject to tampering or misrouting. By encrypting at the source only and not decrypting until the communication reaches its ultimate destination the information content of the message is only usable by recipients who possess the appropriate key. This technique is known as end-to-end encryption, and requires that the message header which contains routing, priority and other information used by the network itself be kept in clear text. In this case, the data security device must be sensitive to the data communication procedures used in the network or be capable of detecting START ENCRYPTION/STOP ENCRYPTION instructions in the text.

DATA COMMUNICATION SECURITY

TECHNIQUE / PROTECTION PROVIDED	LINK ENCRYPTION	END-TO-END AUTHENTICATION	END-TO-END SELECTIVE ENCRYPTION/AUTHENTICATION
MESSAGE SECRECY	X		X
MESSAGE INTEGRITY	X	X	X

Figure I

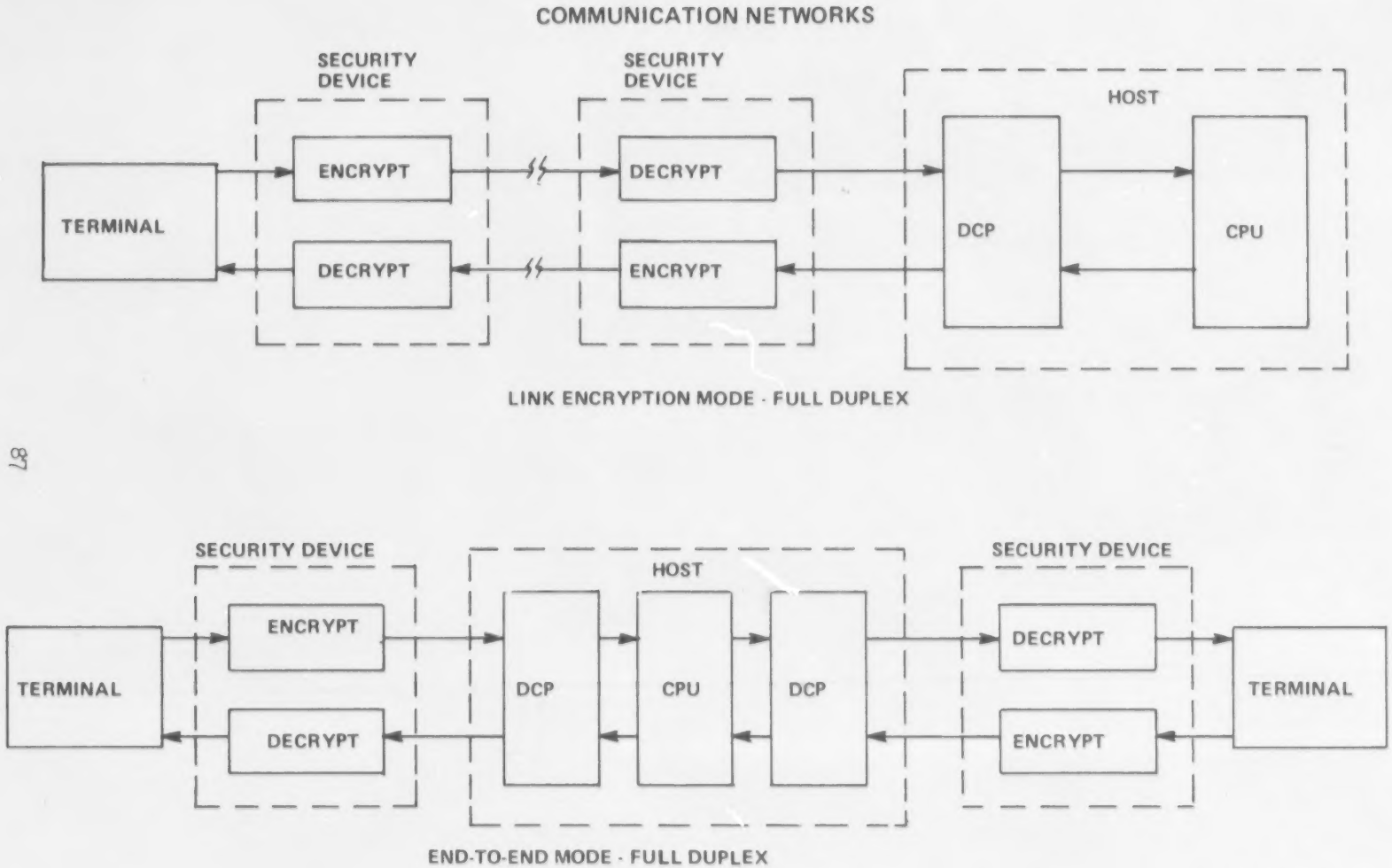
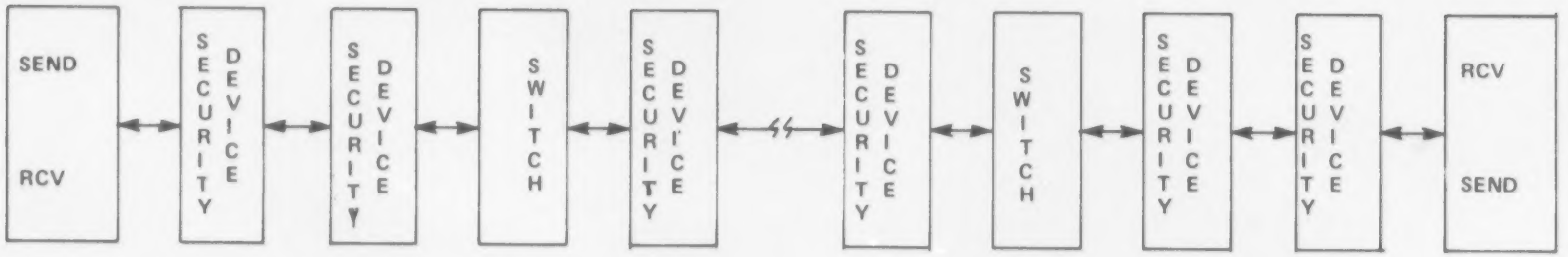

COMMUNICATION NETWORKS
SECURITY DEVICE
SECURITY DEVICE
HOST
TERMINAL
ENCRYPT
DECRYPT
DECRYPT
ENCRYPT
DCP
CPU
LINK ENCRYPTION MODE - FULL DUPLEX
SECURITY DEVICE
HOST
SECURITY DEVICE
TERMINAL
ENCRYPT
DECRYPT
DCP
CPU
DCP
DECRYPT
ENCRYPT
TERMINAL
END-TO-END MODE - FULL DUPLEX

Figure IIa

COMMUNICATION NETWORKS

LINK TO LINK ENCRYPTION



88

END TO END AUTHENTICATION OR ENCRYPTION

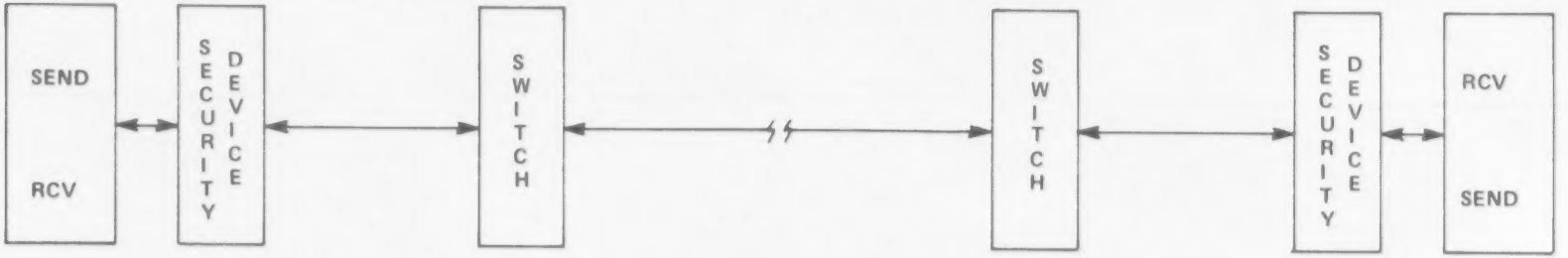


Figure IIb

93

The end-to-end mode is applicable in systems where no operations need be performed on the encrypted data but the data must be transmitted through a switching network and its privacy safeguarded. An example of this could be one IRS branch retrieving tax information on an individual and forwarding it to a second branch, or the banking community accessing credit ratings.

On the other hand, link encryption may be more desirable in the field of International Electronic Funds Transfer where the volume of traffic and/or the message sources and destinations need to be concealed.

However, you can see that the employment of link encryption will be more costly since an encryption device is required at each node rather than just at the source and the destination.

Therefore, a careful analysis of the user's environment and requirements will dictate which mode of operation will yield the level of security desired in the most cost effective way.

Noting the modes of operation, we may now look into the implementation aspects of the DES.

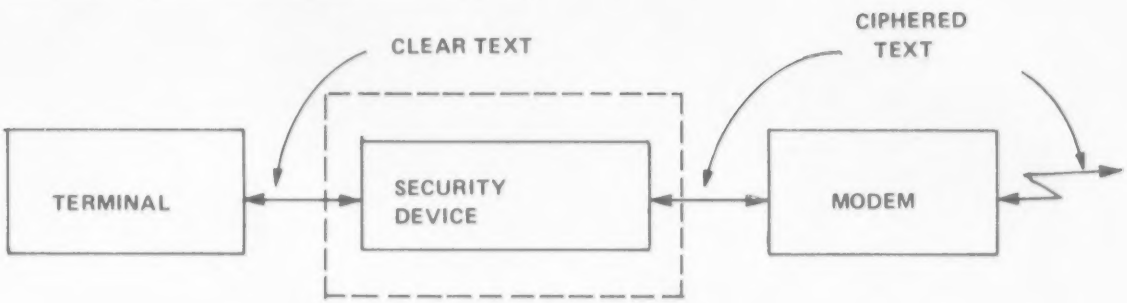
This algorithm can be implemented in a number of ways depending on the user's requirements. It can be used in the encryption mode or the authentication mode; it can be incorporated as an in-built feature of terminals or modems or operate as drop-in, stand alone equipment. This is illustrated by Figure III.

Looking at the advantages and disadvantages of built-in versus drop-in implementation it can be said that in the area of access prevention the built-in implementation is superior. This technique reduces the chance that detection can take place between the terminal and the security device where the text is in the clear. However, this technique may be difficult to implement in existing systems and could require major redesigns. Herein lies the advantage of a stand alone unit which can simply be inserted into existing networks with little or no impact to extant equipment.

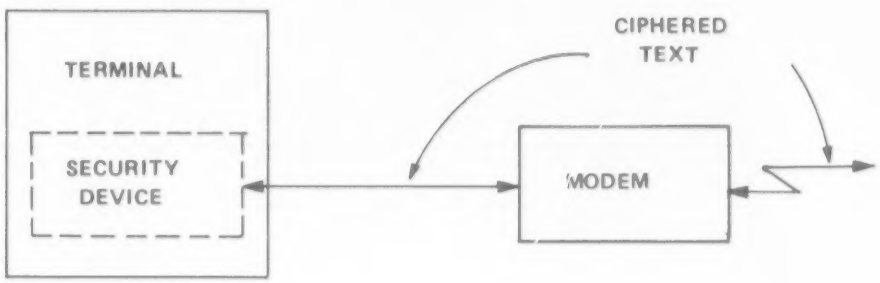
Due to the myriad of user environments and requirements there will be a proliferation of security devices in the marketplace; and indeed, one can see the need for imposing standards on the application of the DES so that the impact on existing networks can be minimized, since the DES is but a <u>part</u> of the Security Device.

VARIOUS METHODS OF IMPLEMENTING DES IN DATA COMMUNICATIONS NETWORK

I. DROP-IN DEVICE



II. IN-BUILT IN TERMINAL



III. IN-BUILT IN MODEM

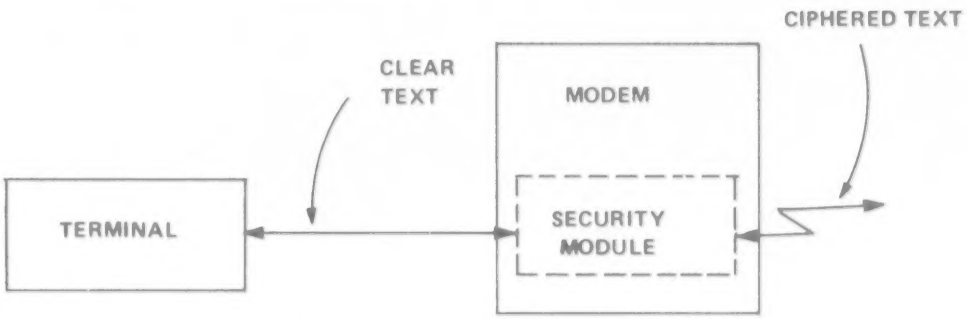


Figure III

90

Areas where standardization is required in the application (implementation) of the Data Encryption Standard are:

- definition of pin-outs - several of the IC manufacturers have undertaken to implement the DES using LSI technology. It is imperative that these packages be standardized as far as the assignment of input/output signals and voltages to a common pin configuration. This would permit interchangeability of the DES device while permitting flexibility in the application of the device.

- power dissipation - particularly with in-built modules where an excessive power drain could have an impact on existing equipment.

- key management - the ability of the user to change the key without the manufacturer's involvement. This area alone is deserving of a great deal of attention. It involves the generation of the key code book and key assignments to correspondent users; the physical protection of the keys. The methodology of changing the keys and the impact of this change warrant investigation. Should the keys be generated by a computer? How often should they be changed? What is the procedure if the keys are compromised? These and many other questions should be resolved in the near future by a key management standards committee.

- data rates - at the present time 56K bits per second is the fastest rate at which data is transferred in data communications networks, excluding multiplexing. However, in the foreseeable future transfer rates may increase and depending upon clock rates and loading and unloading schemes we may find the algorithm processing time, or throughput, reaches a limit. This area should be analyzed and limits (or standards) placed on the parameters which would affect the DES operation.

- data communications interfaces - between the DTE (Data Terminal Equipment) and security devices should be carefully specified particularly in light of new and proposed Federal Standards as well as those of industry.

  Table 1 illustrates the number of interrelated standards for data communication system interfaces. Standardizing this interface will enable security device manufacturers to produce components and devices with greater assurance of wide applicability while assuring functional interchangeability.

  Various committees including ANSI X3 committee, as a result of a request of the IEEE and the NBS/NCS program within the Federal Government are starting to work to structure this standard. Burroughs has proposed an interface standard project to ANSI X3/SPARC committee for consideration.

FEDERAL STANDARD	ASSOCIATED INTERFACE STANDARDS				
	MIL-STD	EIA	ANSI	CCITT	ISO
1020	188-114	RS-422		X.27, V.11	
1030	188-114	RS-423		X.26, V.10	
Proposed 1031		Proposed RS-XYZ		V.24, V.10	DP4902
Proposed 1029		Proposed RS-ABC		V.24, V.10, V.11	DP4902
Proposed 1040			Proposed "ANSI X.21"	X.21, X.24	DP4903
		RS-232C		V.24, V.28	DP2110

TABLE 1

Summary of Associated Standards

(FROM NCS TIB 76-1)

- initialization procedures - methods for the initial loading of the key and starting the algorithm process.
- resynchronization - standards are needed prescribing the methods for resynchronizing the encryption process when synchronization is lost due to power transients or transmission errors.
- levels of security provided by the user - if the physical security of the user's facility is maximum then the security device need not have in-built protection devices. However, where physical security is easily compromised the key storage should be such that upon unauthorized access the key will be destroyed.

- error detection - when the integrity of the message is lost through fraudulent alteration a prescribed alarm should be given. However, for parity errors or errors caused by loss of synchronization a different alarm should be raised.

Each of these salient points requires an exhaustive study as it applies to the use of the DES in order to assure the users that they have equipment which will deliver the desired level of security.

While there are a finite number of applications known to date, there will be many new ways to employ this powerful tool. Before these new methods are applied they must be carefully scrutinized to determine what, if any, impact will result in the data communication community. It is conceivable that as major computer switching networks become interconnected the innocent introduction of any non-standard element into the system could cause great confusion and prevent system operation.

Some points to consider in any application are:

- Strive for transparency.
- Key storage should be non-volatile except upon tampering by unauthorized personnel.
- Key entry should be uncomplicated.
- Universal applicability is more desirable than special purpose equipment.

Encryption is the time-honored way to keep data safe and secret. The DES algorithm now offers a standardized tool to government and the private business sector which through proper use affords the necessary level of security to meet the new regulations on privacy. It remains with us to standardize its applications for the mutual benefit of the entire community.

Integrated System Design

Dr. Walter Tuchman
International Business Machines Corporation
Kingston Development Lab
Neighborhood Road, D69L
Kingston, New York 12493

The following paper has been extracted from the verbal presentation of Dr. Tuchman at the February 15th Conference. A written paper had not been submitted at the time of publication of these proceedings.

1. Introduction

I also have observed during the presentations today that many of the speakers have covered some of the topics that I wish to discuss. However, I would like to take a deeper look into the system architecture required to support encryption. In particular, I would like to talk about the facilities that are required to generate and distribute the encryption keys required by the Data Encryption Standard. My remarks should be taken as a tutorial as I will not be discussing any particular product offering. I will be discussing an integrated approach to the DES algorithm and talking about some of its pros and cons as contrasted to a non-integrated approach.

2. Implementing the DES at the Terminal

The DES can now be readily implemented in LSI for use to computer terminals. The entire implementation of the DES and its necessary control logic can be implemented on a single card and located in a terminal. A throughput of one million bits-per-second can be achieved in this approach.

For the integrated approach, I am assuming that a message packet communications system is available between a terminal and the central processing unit (CPU). The data to be protected is carefully delineated from the addressing and control information in the packet. In this approach, where only the data is encrypted, intermediary nodes in the network need not have an encryption capability nor even know that the data is encrypted. With this approach, performance and security are improved and the cost is minimized.

A vulnerability of data is actually being designed into the newer computer network architectures. This vulnerability is especially prevalent in a loop network. Data from all terminals co-exists on a

common communication line and every terminal has the capability of reading the traffic passing through the line. Encryption of the data in an end-to-end security network offers a unique and cost effective solution to this problem.

If we contrast the end-to-end approach with the simpler approach of link encryption, which I call line bracketing, we find some interesting comparisons. First, the line bracketing approach can be implemented quite simply and will provide security on a simple communication line. Line bracketing boxes have very little degradation, typically use all codes, can be used on practically all line disciplines and can become very nearly a universal box for use between any modem and terminal. We probably will see the integrated approach and the line bracketing approach used concurrently for the foreseeable future.

Some of the disadvantages of the line bracketing process are readily apparent, i.e., in dial-up networks there is no key management service, and the keys in many devices must all be the same and must be manually changed. Line bracketing units usually encrypt the control information as well as the data and hence, cannot be used either in loop applications or in most packet-oriented networks.

3. Implementing the DES at the CPU

There are three different ways of implementing the DES at the CPU: locating the DES device in the front-end communications processor, locating it in the channel within the CPU or locating it in a hardware device controlled by the CPU. The advantage of the first is that the identical DES device may be used in the front end processor that is used in the terminal. The DES in a channel requires very high speed capabilities, perhaps 50 million bits-per-second throughput. The CPU implementation requires, as does the channel implementation, a multi-chip DES for high speed reasons.

The integrated approach of implementing the DES as a CPU hardware device requires a very careful solution to the key management problem. With that in mind, I will define what I call an "optimum" solution to key management and key distribution in an integrated CPU and DES facility.*

Let us design a network consisting of N devices attached to a CPU. Each terminal has an imbedded, private Device Key (the encryption key to be used with the DES). Each key is different for good security. The question is, "How can any device talk to any other device if all of the keys are different?" The solution is to maintain a list of all of the private Device Keys in the memory of the CPU and let the CPU generate a new key for use in protecting the data between any two common devices.

---

*Editor's Note: Dr. Tuchman told a lengthy, humorous story at this point to illustrate his definition of the word "optimum."

To prevent this list of keys from being stolen or accidentally lost, we will encrypt this list of keys with another key which we call the Master Key. This key is located only on the DES device and cannot be read by anyone.

The following happens during a "session" of communications between any two of the devices. Let us say that terminal 2 wants to talk to terminal 8. The private keys for terminals 2 and 8 are both contained in the encryption key list which, of course, is encrypted by the Master Key. The CPU generates an encrypted Session Key from a device that is time-dependent and pseudo-random, such as the system clock. This encrypted Session Key (defined to be encrypted under the Master Key and never appearing in the CPU in its plain form) as well as the Private Key for terminal 2 and the Private Key for terminal 8, are all sent to the DES device controlled by the CPU. The Session Key is decrypted using the Master Key; the encrypted Device Key for terminal 2 is decrypted using the Master Key; and then the Session Key is encrypted using the Device Key of terminal 2. Similarly, the encrypted Device Key of terminal 8 is decrypted using the Master Key and the Session Key is encrypted using it. The encrypted Session Key is then sent to terminal 2 protected by the Device Key of terminal 2 and the encrypted Session Key is sent to terminal 8 protected by the Device Key of terminal 8. The Session Key is then decrypted at terminal 2 and at terminal 8 using their respective Device Keys. Thus, both terminal 2 and terminal 8 have the same Session Key and will be able to communicate.

That is the "optimum" solution we have found for key management in an integrated system design.

An LSI Implementation of the Data Encryption Standard

Howard O. Wright
Rockwell International
Mail Station 503-200
4311 Jamboree Road
Newport Beach, CA 92663

This paper describes an LSI circuit designed to perform data encryption using the algorithm adopted by the National Bureau of Standards as the Data Encryption Standard. The encryption unit enciphers/deciphers data in 64-bit blocks. Both input data and output data are buffered, allowing the unit to sustain a data rate up to 1.6 Mb/s. The unit has tri-state busing capability and is a versatile LSI unit, designed for use in a wide variety of applications. The unit is sufficiently small in size and low in power consumption and cost that it will allow data encipherment to be used in systems in which the use of encipherment was previously economically unfeasible.

Key words: Encryption; LSI; MOS; security.

1. Introduction

Collins Radio Group of Rockwell International Corporation has implemented an MOS circuit that is designed to perform the algorithm designated by the National Bureau of Standards as the Data Encryption Standard (DES).1/ The 64-bit block enciphering system described herein consists of a method of enciphering or deciphering a 64-bit block of input data into a 64-bit block of output data with a variable, 56-bit key.

A single, 40-pin MOS circuit is described herein that performs the algorithm function and accomplishes a task that previously required over 100 medium scale integration (MSI) circuits to implement. The purpose of this paper is to describe a large scale integrated (LSI) circuit implementation of this algorithm. Rationale for the implementation and some of the ways envisioned for the circuit use will be discussed.

DATA PATH BLOCK DIAGRAM

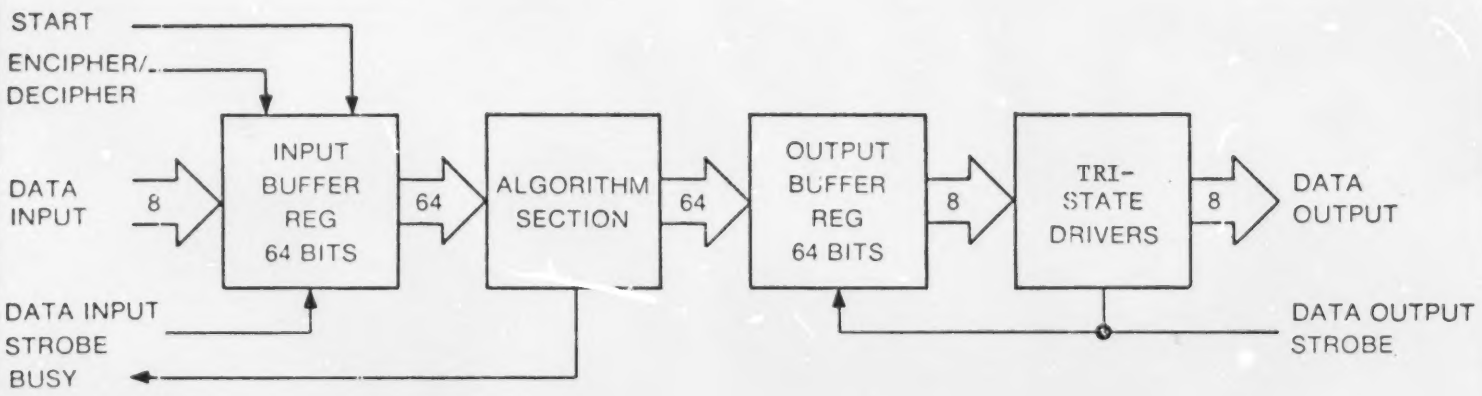


Figure 1. Data Path Block Diagram.

2. Architecture

The architectural design of the LSI Encryption unit was influenced by a number of application and technical parameters. The largest anticipated application for the units is in the terminal field. Many terminals are being designed around 8-bit, character-oriented microprocessors; therefore, architectural design was directed toward an 8-bit parallel input/output (I/O) terminal with busing capability. The design was also influenced by initial customer requirements for a unit with a throughput capability of at least 1 Mb/s.

The basic data path architecture for the LSI encryption unit is shown in figure 1. The data enciphered/deciphered are loaded into a 64-bit input buffer register, 8 bits at a time, using the data input strobe line to control the transfer. Eight data input strobe pulses are required to complete the load of the 64-bit input buffer. The 64-bit input register is implemented using eight 8-bit shift registers. The initial permutation defined by the DES is accomplished at the input register by connecting the 8-bit input to the shift registers. Following a load of eight 8-bit bytes of data, the contents of the input register will be as defined by the DES initial permutation table.

Once the input buffer has been loaded, the start line will be pulsed to initiate operation of the algorithm section. The start pulse causes the contents of the input buffer register to be transferred to the algorithm section, and frees the input buffer register to receive another block of data. One pin on the unit is used to specify whether the processing is to encipher or decipher the input message. In either case, I/O is identical.

To allow the unit to sustain a data rate up to 1.6-Mb/s in a pipelined mode of operation, 64-bit buffer registers are used on both input and output. This type of architectural design allows simultaneous data input, algorithm unit processing, and data output.

The output buffer is a 64-bit buffer that is organized as eight 8-bit shift registers. The inverse of the initial permutation is accomplished at the output buffer in the same manner as the initial permutation was accomplished at the input register. A data output strobe line is used to transfer 1 bit from each of the eight registers to the output pins of the unit, and the data in each of the output buffer shift registers is shifted down 1 bit on the falling edge of the data output strobe. Eight data output strobe pulses are required to extract the contents of the output buffer register.

Two additional control signals are required to use the unit in the pipelined mode of operation: one is the busy signal, and the other is the enable output buffer load signal.

When the algorithm section is processing data or when it is holding previously processed data while waiting to load the output buffer, a busy condition is indicated by the busy signal. The busy signal goes high following a start pulse, and remains high until the algorithm section has transferred a block of 64 bits to the output register. The falling edge of the busy signal is an indication to the external control logic that data is available in the output buffer and that the start line can be pulsed to start a new processing sequence.

The enable output buffer load signal line is pulsed to indicate to the algorithm section that the output buffer can be loaded when data is available. For normal block enciphering, this line will be pulsed after the eighth data output strobe has emptied the output buffer. Although the buffer empty signal could be generated internal to the chip by counting eight data output strobe pulses, there are cases in which only a few of the 64 output bits are actually used; therefore, allowing the external logic to determine when the output buffer can be loaded increases unit versatility.

2.1 Key Variables

The key variable used by the algorithm section is stored in an internal 56-bit key register. The method of handling key variables during load of the key register was influenced by the requirement in some systems to ensure complete physical separation between key variables and normal data paths. Consequently, eight pins on the package were devoted to a clear key port for use in entering clear key variables into the unit, as shown in figure 2.

A key strobe is used to enter key variable data, 8 bits at a time, into the unit through the clear key port inputs. Each 8-bit group is checked for odd parity as it is entered. Following removal of the parity bit, 7 bits of key variable information are entered into the key register each time the key strobe line is pulsed. A parity error line is available on an output pin and will be set if odd parity is not present on the clear key port while the key strobe line is high.

In addition to having the capability of loading a clear key through the clear key port, as described, the unit has a provision for deciphering key variable data in the algorithm section, checking the resulting clear key data for correct parity, and transferring the result to the key register. This process allows keys to be entered into the system in enciphered form. The enciphered keys can be carried to the unit by courier or can be transmitted to the unit via the normal communications path.

When enciphered key variable data are entered into the input buffer over the normal input data lines, the rekey line is pulsed instead of the start line to start the algorithm. When the algorithm unit has completed deciphering the key variable, it is loaded into the output buffer, checked for proper parity, and finally loaded into the key register. During this

KEY PATH BLOCK DIAGRAM

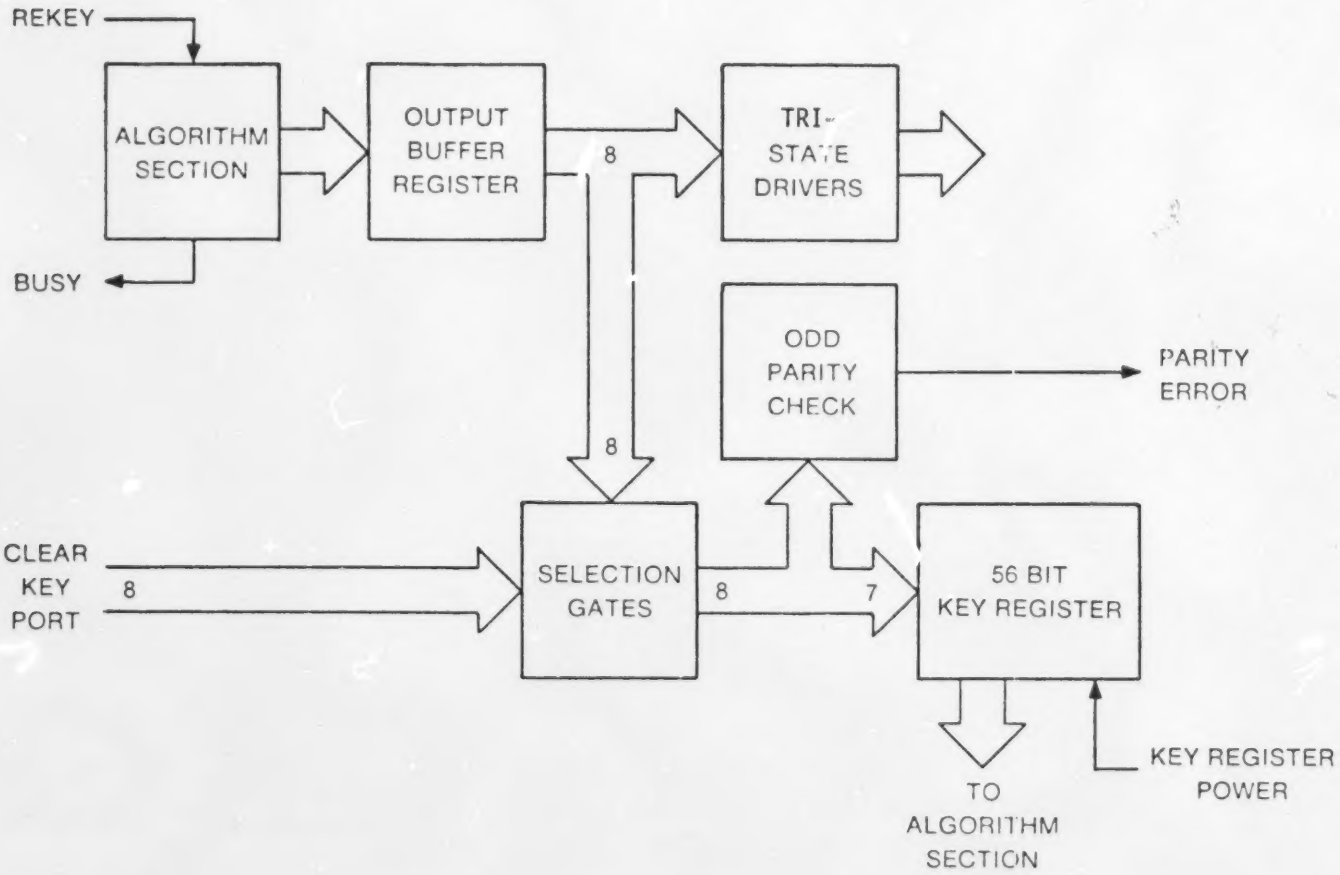


Figure 2. Key Path Block Diagram.

process, the tri-state drivers on the output are forced off to prevent clear key variable data from being observed on the output.

An important security precaution is taken during the rekey process by gating off the tri-state output drivers and clearing the output buffer following the rekey process. This feature, when coupled with a suitable key distribution procedure, can help prevent a covert attempt to obtain key variable information from the unit even though an intruder has access to the unit's circuitry. This is particularly important at an unsecured remote terminal site.

The capability to power the key register through a separate power pin, which allows a small battery backup to be designed into the system to provide nonvolatile key storage (lasting up to several days during power outage condition), is an additional feature designed for use at a remote terminal.

2.2 I/O Busing

To facilitate using the unit in systems built around a bus concept, two signals were added that allow the control signals previously described to be bused. A control enable signal was ANDed with the start, rekey, encipher/decipher and the enable output buffer load lines. The control enable signal is designed to be connected to the addressing function associated with the bus; and, when a bus output sequence is addressed to the control inputs of the algorithm unit, the control enable signal is momentarily raised to gate the state of the four control signals from the bus into storage elements within the unit.

The busy and the parity error signals are gated to the output pins of the unit through tri-state drivers. The drivers are enabled by a status enable signal that allows the busy and the parity error signals to be gated onto the bus when addressed by the bus addressing function. For systems that do not employ busing, the control enable and the status enable signals can be tied to a logical state and ignored.

2.3 Algorithm Section

The block diagram shown in figure 3 illustrates the complete encryption unit including the algorithm section. As previously described, the initial permutation (IP) is performed at the input buffer. Data are transferred from the input buffer to the 32-bit registers L and R to begin a processing cycle. A processing cycle, either encipher or decipher, is accomplished in 16 processing iterations. As defined by the E bit selection table in the DES, 48 bits are selected from the R register, and are EXCLUSIVE ORed with selected bits from the key register as defined by Table PC-2 of the DES. The resulting output is used to address Read Only Memories (ROM's) S1 through S8. Output bits from the ROM's are selected according to the primitive function P, and are exclusive ORed with the 32-bit contents of the L register. The final step in an iteration process is to transfer the contents of the R register to the L register, and to transfer the results of the last exclusive OR into the R register.

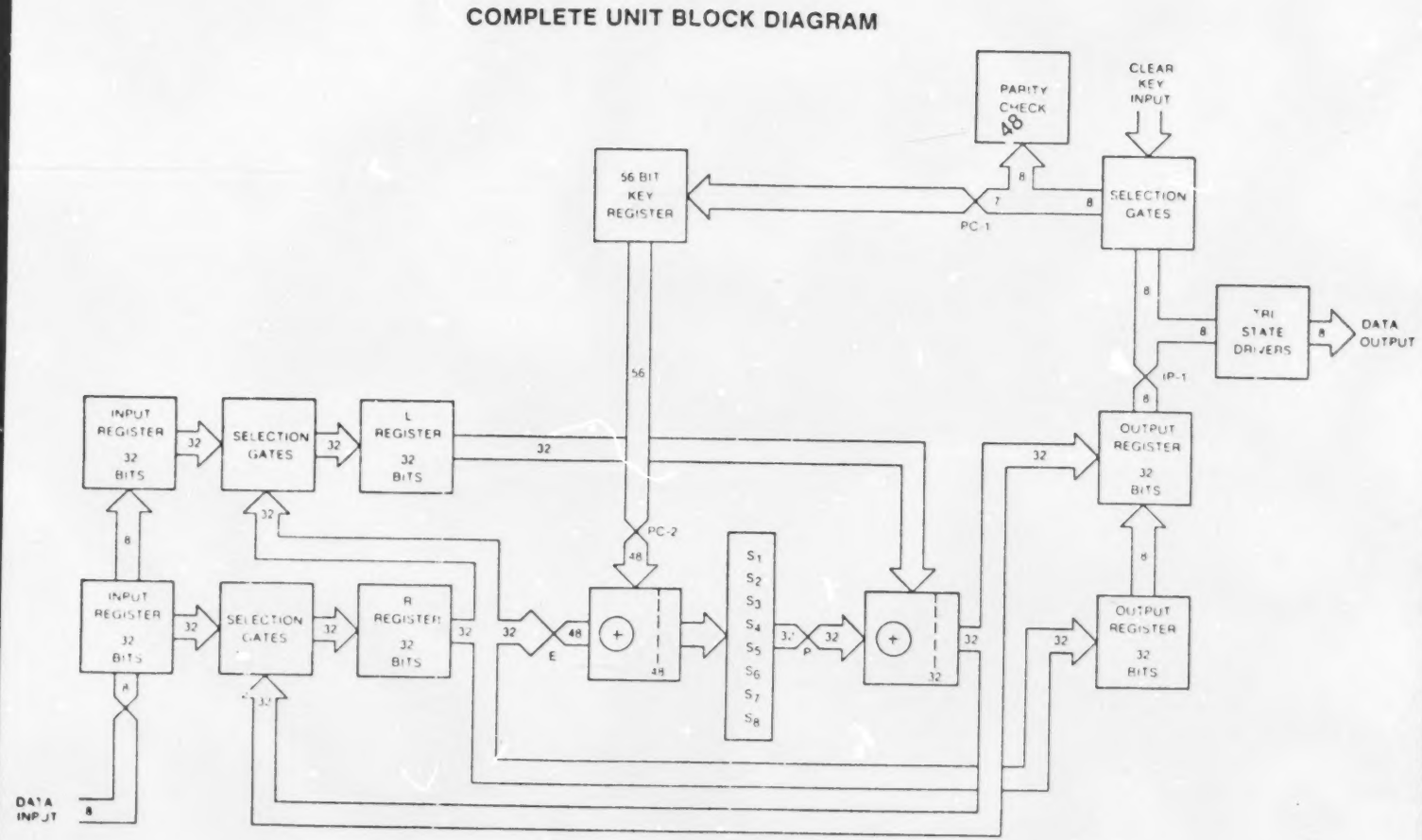


Figure 3. Complete Unit Block Diagram.

Between iterations, the key data are rotated in the key registers, as defined by the DES. Following the sixteenth iteration, data are transferred to the output buffer register; and the inverse permutation (IP^{-1}) is accomplished at the output of the output buffer registers, as previously described.

2.4 Final Product

The circuit is built using PMOS technology, and is contained in a 40-pin package. The unit requires +5 and -12 volts and dissipates 300 MW of power. A free-running clock is required to run the algorithm section. The time to process a 64-bit block of data is dependent upon the speed of the clock, and is defined by:

Time = (Period Clock) (64)

The maximum clock frequency cannot exceed 1.6 MHz.

3. Applications

Figure 4 presents an example of how the unit can be used in a microprocessor system. The unit is completely under the control of the processor. The processor loads the unit with 64 bits of data to be enciphered or deciphered, starts the unit, and then reads out the 64-bit result. For applications requiring higher throughput, more than one unit can be connected to the bus; also, the unit could be connected to the bus through a DMA channel to relieve the processor of handling each byte of data.

There are many applications in which the requirement to handle enciphered data in blocks of 64 bits is too restrictive (i.e. the case of an interactive terminal connected to a processor). For such applications, the algorithm unit was designed to support a cipher feedback system. In this mode of operation, data are enciphered by EXCLUSIVE ORing it with the output of the algorithm unit. The enciphered data are then loaded back into the algorithm unit and the algorithm unit is started. After the algorithm unit has loaded its output register, the next clear text data are enciphered, and the cycle repeats. Only 8 bits of the 64 bits generated each cycle at the output of the algorithm unit are used, and the unit is cycled after only 8 bits have been loaded into the input register. The input to the algorithm for each cycle then becomes the previous 64 bits of enciphered data.

This system has the advantage that, once eight characters have been passed through the system to synchronize the receiving algorithm unit, a character will be deciphered at the receiver for each character input at the transmitting end. Therefore, the block encipherment requirement is eliminated.

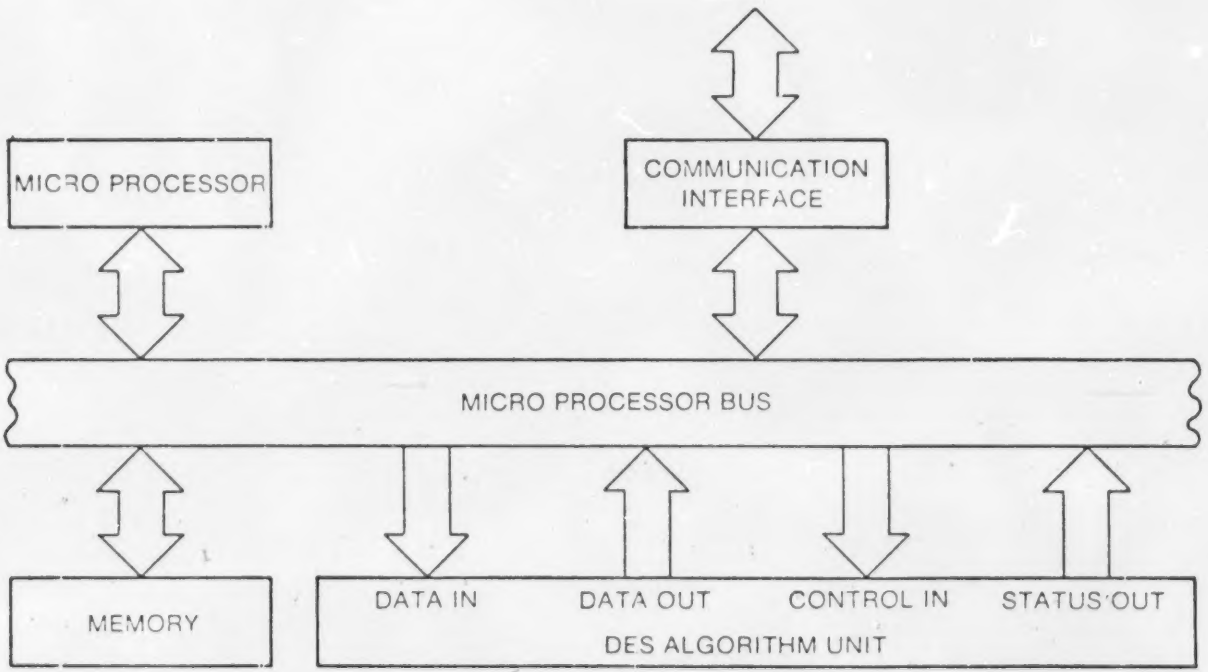


Figure 4. Microprocessor Based System.

Similar techniques can be applied to handle serial data or data in any character width from 1 to 64 bits; however, when data path widths other than multiples of eight are used, an accumulating register at the input to the algorithm units is required.

4. Conclusion

The 64-bit block enciphering circuit is a versatile LSI unit with high throughput capability that was designed for use in a wide variety of applications. The unit defined herein provides the system designer with a powerful and cost-effective tool for solving many of the data security problems that currently face the industry. The unit is sufficiently small in size, and low in power consumption and cost, that it will allow data encipherment to be used in systems in which the use of encipherment was previously economically unfeasible.

5. References

1/ "Data Encryption Standard," Federal Information Processing Standard Publication 46, National Bureau of Standards, January 15, 1977.

A Microprocessor Controlled LSI Implementation of the Data Encryption Standard

Keith Warble
Motorola Inc.
Government Electronics Division
Scottsdale, Arizona

and

Durrell Hillis
Motorola Inc.
Government Electronics Division
Mail Station 2289
8201 E. McDowell Rd.
Scottsdale, Arizona 85252
Telephone: (602) 949-4735

Presented is an LSI implementation of the Data Encryption Standard. The device has been developed for use with microcomputer based data processors, with encryption or decryption of 64 bit blocks inputted and outputted through a single 8 bit tri-state bus port. A single +5 volt supply powers the LSI chip; block processing time is 160 microseconds, allowing typical MPU configurations to operate over 200 Kb/s. The unit possesses two key registers to facilitate downline loading of encrypted key, with on-chip decryption and error checking under the control of a resident master key. Continual checking of the operating key during algorithm execution as well as during key load provides an economical degree of security for many applications.

Key words: Communication; encryption; LSI; microprocessor; MOS.

1. Introduction

A hardware LSI implementation of the Data Encryption Standard has been developed at Motorola Government Electronics Division. The device, called the Data Security Device (DSD), performs the 64 bit block encryption or decryption according to the Federal Standard algorithm, utilizing one of two 56 bit keys stored on chip. Plain and cipher data blocks are inputted and outputted through a single 8 bit tri-state I/O port, so that minimum load is presented to a microprocessor data bus.

The DSD is configured on a Silicon Gate N-Channel MOS Depletion Load LSI chip contained in a 24 pin package to minimize device cost.

This paper will discuss the flexible control features of the chip design, and applications of these features in secure data module implementations.

2. Data Security Device Construction

The Data Security Device was designed to provide the DES security function for many existing terminal, link and computer systems. Since a large number of these systems now utilize microcomputer devices for processing and formatting of data, emphasis has been placed upon the ease of implementing security with the DSD chip in existing microprocessor hardware. Use of a single 5 volt power source, conventional clock sources, and minimum data bus loading are features of the DSD.

3. DSD Architecture

The Data Security Device appears to an MPU system as an Interface Adapter device. An illustrative example of such a system, with the encryption function added, is shown in figure 1.

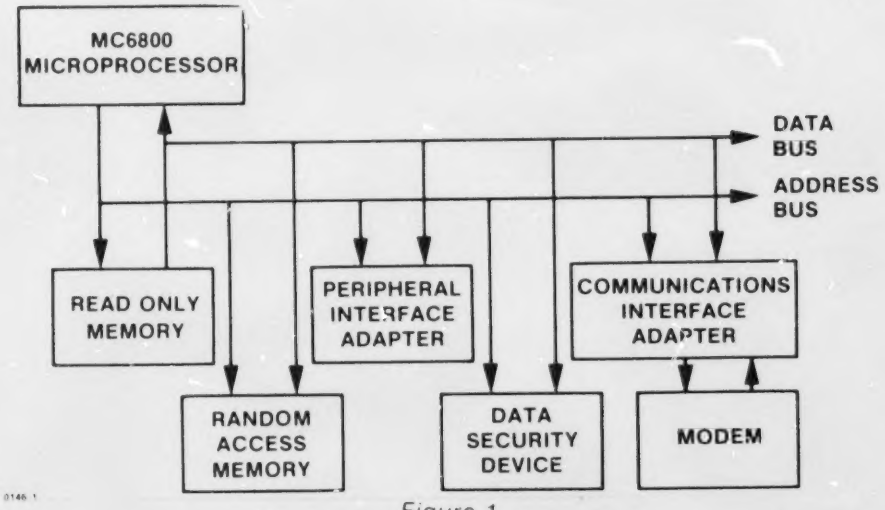


Figure 1

M6800 MICROCOMPUTER FAMILY BLOCK DIAGRAM

Internal construction of the DSD is illustrated by the block diagram of figure 2. The device consists of a single 8 bit Data Bus Buffer with tri-state operation, through which data may be entered into 64 bit Active or Major Key Registers or a 64 bit Data Block Register. Output data from a Status Register or the Data Block Register is also switched through the Data Bus Buffer.

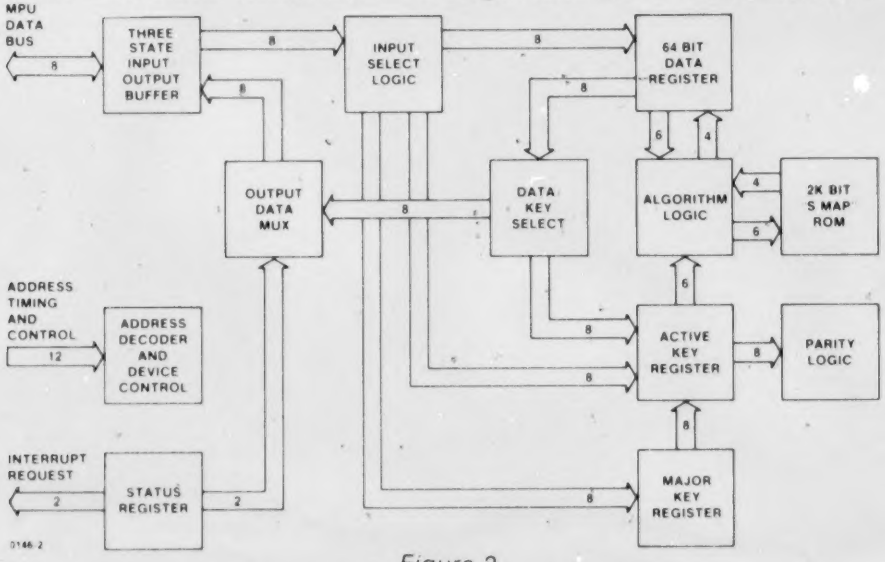


Figure 2

DATA SECURITY DEVICE BLOCK DIAGRAM
(DES ALGORITHM)

At the bus interface, the Data Security Device (DSD) appears as eight addressable memory locations to the MPU, through which the operational mode of the chip may be selected, chip status monitored, key or data written into the device, and data read from the device.

As shown in table 1, the operation of the DSD is split into five major modes: (1) Data Encryption, (2) Data Decryption, (3) Loading of Data or Encrypted Key, (4) Data Readout and (5) Status Readout. These and additional control modes are activated by three address input lines and a Read/Write input command.

CONTROL ADDRESS				OPERATIONAL MODE
A0	A1	A2	R/W	
0	0	0	0	WRITE DATA/"C" KEY OPERATION
1	0	1	0	ENCIPHER DATA
0	0	1	0	DECIPHER DATA
0	0	1	1	READ DATA
1	0	0	1	READ STATUS

* Instruction performed during eighth byte of Data Block entry.

0146-4

Table 1.

MAJOR OPERATIONAL MODES OF DATA SECURITY DEVICE.

Table 2 illustrates additional control operations which initialize the chip and determine the operational key to be used. Since the writing of ciphered key appears as data to be processed, the control address present at the eighth byte of data block entry is used to determine whether the processed data can be made available for output (valid data) or loaded into the Active Key Register.

	CONTROL ADDRESS				CONTROL MODE
	A0	A1	A2	R/W	
	1	0	0	0	RESET/INITIALIZE
	0	1	0	1	ACTIVATE MAJOR KEY
	1	1	0	0	ACTIVATE PLAIN SECONDARY KEY
*	1	1	1	0	DECIPHER SECONDARY KEY
*	0	1	1	0	ENCIPHER SECONDARY KEY

* Instruction performed during eighth byte of Key Block entry.

0146 5

Table 2.

CONTROL MODES OF DATA SECURITY DEVICE.

4. Chip Initialization

A $\overline{\text{RESET}}$ signal input to the DSD is used to initialize the internal control logic, status flags, and counters. The $\overline{\text{RESET}}$ function should be coupled with the system power on reset to provide orderly system initialization and also may be used as a master reset to the chip during system operation.

Reinitialization may also be performed under software control by a write command under address control A0 = 1, A1 = 0, A2 = 0, R/W = 0.

5. Key Operations

Two key registers in the DSD allow storage of a Major Key while processing data with an Active Key. Both key registers are loaded through the data bus port, with command addressing dependent on the form and destination of the key.

The prime or master key is entered into the Major Key register and simultaneously checked for parity error and loaded into the Active Key Register. During algorithm operation, the DSD continually performs parity checking on the contents of the Active Key Register.

A secondary key may be loaded into the Active Key Register in plain or ciphered form. If the secondary key load command shows a cipher key operation, the DSD will

process the key using the present Active Key. The DSD must have previously been loaded with either Major Key or another Secondary Key. After algorithm processing, the DSD transfers the deciphered Secondary Key to Active Key Register while checking parity. Should the Secondary Key contain parity errors, as is possible with down line loaded data, a repeat cipher key operation may be performed using a Major Key transfer. During Secondary Key or transfer operations, the contents of the Major Key Register are preserved.

6. Enciphering of Data

For the enciphering process to take place a key, major or secondary, must be resident in the Active Key Register. Data is written into the device in eight eight-bit bytes under software control. The first seven bytes are written into the device under address control A0 = 0, A1 = 0, A2 = 0, R/W = 0. The eighth byte is written under address control A0 = 1, A1 = 0, A2 = 1, R/W = 0. After the eighth byte has been written, enciphering of the Data block automatically commences utilizing the key stored in the Active Key Register.

As the enciphering algorithm is initiated, the key is checked for parity error, which if detected, sets the Key Parity Error flag. Any external action other than a read request of status (A0 = 1, A1 = 0, A2 = 0, R/W = 1) during the actual enciphering process will be ignored by the device.

At the completion of the enciphering process, the enciphered data may be read from the device under software control. For some system applications, e.g., cipher feedback operation, it may be desirable to enter a new block of data without reading out the total block previously enciphered. Input of new data without total readout is therefore not precluded by the DSD.

7. Deciphering of Data

The process of deciphering of data is operationally the same as the enciphering process with the exception that the eighth byte of data is written into the device under address control A0 = 0, A1 = 0, A2 = 1, R/W = 0.

8. Reading of Data and Status

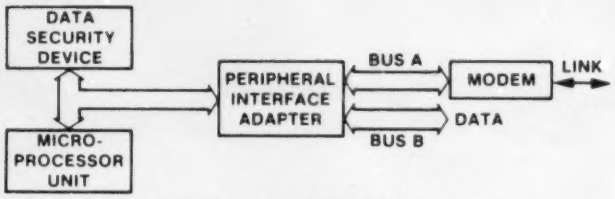
Data may be read from the device in eight-bit bytes under address control A0 = 0, A1 = 0, A2 = 1, R/W = 1. Any attempt to read data while the device is "busy" will be ignored.

Two device status bits are provided which can be read from the device under software control (A0 = 1, A1 = 0, A2 = 0, R/W = 1). Key Parity Error (PE) appears on bus data line D0, and Device Busy appears on bus data line D1. D2 through D7 are held to logic 0 during a read of status. PE and BUSY are also provided in complement form as "open drain" discrete outputs from the device as $\overline{IRQA}$ and $\overline{IRQB}$ for use as interrupt requests and/or status display.

9. Device Operating Configurations

The DSD is packaged in a 24 pin Dual In-Line package. In addition to Data, Address and Status Interrupt pins, six pins are used for Chip Enable and five Chip Select lines, so that several DSD's may be operated under the control of one microprocessor. A free-running 2 MHz clock synchronizes the DSD with MPU configurations; for M6800 configurations, the MC 6871 or MC 6875 clock generator provides 1 MHz system clock to the MPU and 2 MHz to the DSD. Under this configuration, a block processing time of 160 microseconds, and typical input/output of 120 microseconds yield a maximum data encryption rate of approximately 230 Kb/s. DSD operating power dissipation averages 450 milliwatts in this configuration.

Figure 3 shows a typical system application of the DSD/MPU configuration operating in the Cipher Feedback (CFB) Mode. A Peripheral Interface Adapter is used to input unciphered data and output cipher data on a byte-by-byte basis. The configuration makes use of the MPU's exclusive OR instruction and the DSD's encrypt and decrypt capability on consecutive operations. Each character or byte of data is enciphered by exclusive ORing with a byte of the last encrypted block from the DSD. The DSD then decrypts the cipher block to recover the previous enciphered data block and updates this block with the new enciphered data byte. Because approximately 400 microseconds are required for each character processed, the data rate in CFB is slowed to 20 Kb/s. A minor modification to the DSD chip address logic is required to perform CFB operation in accordance with NBS Guidelines. However, it may be desirable to allow room for differing versions of CFB to reduce the bit error extension difficulties anticipated for some communications links.



CIPHER FEEDBACK MODE OPERATION

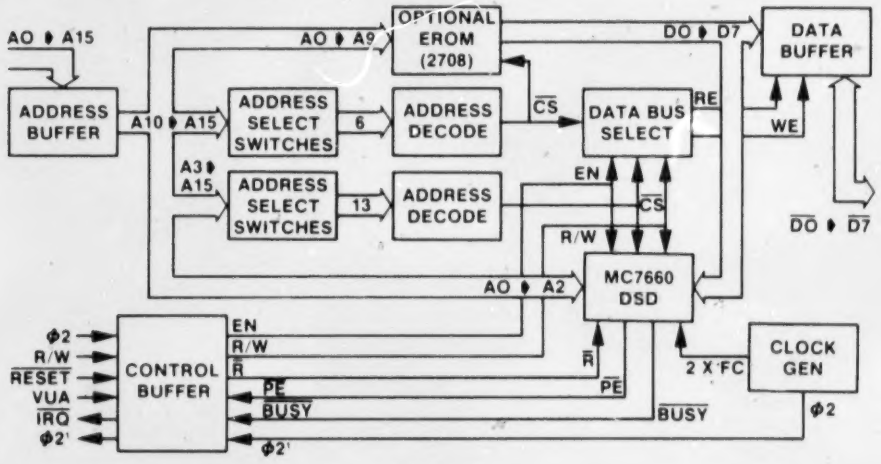
CYCLE	MPU	DSD	PIA A	PIA B
1	FETCH DATA	ENCRYPT LAST BLOCK	-	READ DATA
2	FETCH CIPHER	READ CIPHER BYTE	-	SENSE DATA
3	COMPUTE DATA ⊕ CIPHER	DECRYPT BLOCK	-	STORE DATA
4	READ D ⊕ C	UPDATE BLOCK WITH D ⊕ C	WRITE D ⊕ C	-

0146-7

Figure 3

DATA SECURITY DEVICE SYSTEM APPLICATION WITH CIPHER FEEDBACK OPERATION

The block diagram of a versatile unit for data encryption is shown in figure 4. The unit is configured as a plug in Security Module for the M6800 EXORciser Microcomputer development system. A 9" by 6" board containing the DSD, Address and Data Bus buffering allows the M6800 Security Module to adapt the EXORciser to a secure data terminal. Optional Erasable Read Only Memory in the module can be addressed to load selected encryption keys into the DSD. The Module can be programmed to operate in the Block or CFB cipher mode to provide the EXORciser capability for use as a flexible secure data terminal.



0146-3

Figure 4

MOTOROLA M6800 SECURITY MODULE

10. Conclusion

The Data Security Device, an LSI implementation of the Data Encryption Standard, provides a flexible means of incorporating data security into microprocessor based terminals and minicomputer systems. Design features which make the chip appear as an additional member of a microcomputer family allows economical hardware and software solutions to growing computer security needs.

APPENDIX

Question and Answer Session

The following questions were submitted in writing during the conference. The answers were prepared by either the speaker, the session chairman or the editor.

Question: To Scott Taylor, Collins of Rockwell International

Did you say that the cost would decrease at the same time that the speed and density increase in LSI technology so that we get a factor of 1000 improvement over the next five years?

Answer:

No. You can optimize on any one of the parameters but you cannot optimize on all of them at the same time so that you will get this high a factor of improved LSI efficiency in the next five years. The actual improvement will depend upon whatever factor is being optimized and the cost of optimization.

Question:

How does one obtain a copy of the document referenced by Mr. McDonnell on EFT security?

Answer:

By writing to the National Commission on Electronic Funds Transfer, Washington, D. C. 20429

Question: To Barrie Morgan, Datotek, Incorporated

How do you suppress control characters in the cipher text of a DES device?

Answer:

The algorithm used to suppress forbidden characters depends on the code set being used. ASCII and EBCDIC each require different forms of suppression. In some cases, a look-up table can be used.

Question:

Whom may I contact to obtain more information on DES protocols in communications applications?

Answer:

There are two sources: Dr. Dennis K. Branstad of NBS, phone number 301-921-3861; and Mr. Ed Stephan, GSA, phone number 202-566-1180.

Question:

Would NBS please reconcile the fact that an encryption algorithm capable of being implemented in either hardware or software is needed, but that the DES is only to be implemented in hardware?

Answer:

The DES is only to be implemented in hardware, i.e., electronic devices with read-only memory, micro-programs or in dedicated microprocessors. The DES can be validated easily in these forms and is very difficult to be modified by unauthorized people. Software may be used to interface the DES device to its application. Software algorithms were not considered in our solicitation or our evaluation for these reasons.

Question: To Barrie Morgan, Datotek, Incorporated

If messages consist of digits only (typical of EFT transaction data), is the security provided by DES compromised by enciphering only the least significant bits of an 8-bit code?

Answer:

To the best of our knowledge, the security provided by the DES will not be compromised if only the least significant bits of the data code are enciphered. This technique may be used to assure that control characters do not appear in the cipher stream.

Question: To Stephen Walker, ARPA

In a packet oriented system, such as HDLC, how can encrypted data be routed?

Answer:

End-to-end encryption techniques, applied in packet-switched networks, require that one encrypt the data only. The headers and control

information must be transmitted in the clear. If the address information is sensitive, then link encryption must be used also between the switches.

Question: To Scott Taylor, Collins of Rockwell International

What is the overall effective speed in the Collins implementation of the DES?

Answer:

1.6 megabits. On the Collins chip, input, output and processing are all done in parallel in order to achieve this high throughput.

Question: To Kris Rallapalli, Fairchild Semi-Conductor

What is the approximate cost of the Fairchild four-chip DES device?

Answer:

The estimated price at this time is $200.00.

Question:

What interest has the Government expressed in implementing the DES in military cryptographic equipment?

Answer:

The DES was not designed for use in military applications. This is clearly stated in FIPS PUB 46.

Question:

The Office of Telecommunications Policy Circular 15 significantly downplayed the need for encryption to protect privacy in data communications. What is the position of NBS in response to this question?

Answer:

The Privacy Act of 1974 does not explicitly require encryption. This law does require that an adequate level of protection be provided for sensitive data in high threat environments. Circular 15, as drafted by OTP, simply states that most personal data handled within the Government does not exist in these environments. It does not say that encryption cannot be used.

Question: To Kris Rallapalli, Fairchild Semi-Conductor

How would a DES device encrypt a file on a storage unit such as a magnetic tape so that it is "different" from a similar file on the same unit?

Answer:

Encrypted data is not inherently different from unencrypted data. The DES is totally transparent to data codes and likewise protects all possible data codes. Therefore, a data file must be marked in some way as being encrypted. In addition, the cryptographic protection of stored data requires a different approach from that of communications. First of all, the encryption key used to protect the data must itself be stored and protected as long as the data is to be retained for later use. The key used to encrypt the data must in some way be associated with that file. Methods for achieving this are being developed.

Question: To Keith Warble, Motorola

What is the cost of the M6800 security module that he presented?

Answer:

Our estimated cost for the security module including a DES chip and related interface devices on a 6" x 9" card is $495.00, and will be available some time in mid 1977. (Editor's Note: Motorola has announced an M6800 Data Security Module for $475. and an Intel 8080 Data Security Module for $495.)

Question: To Barrie Morgan, Datotek, Incorporated

Why not process the enciphered data and send the ETX (end of text) in the clear so that single bit errors would be far less likely to disrupt communications?

Answer:

That is the normal procedure. In ASCII the control characters are passed unenciphered and all control characters which happen to occur in the cipher stream are flagged to prevent their being interpreted as control characters.

Question: To Clark Weissman, System Development Corporation

How are automated key management keys protected?

Answer:

A key that is electronically transmitted must be encrypted under a key that has never been transmitted through the electronic network.

Question:

Is the DES under export control?

Answer:

The export of all cryptographic equipment is controlled under Code of Federal Regulations 22: 121-128. The Office of Munitions Control of the United States State Department enforces this regulation. It is expected, however, that licenses can be obtained to export DES devices.

Question: To Carl Campbell, Interbank Card Association

What impact will the DES have on communications networks, i.e., network management and compatibility with existing common carrier networks?

Answer:

The DES can and should be implemented so that it is transparent to network management and has little, if any, impact on the network itself. The common carrier network should not be affected by the DES if it is implemented and used properly. Only a negligible effect will be apparent to users if the communications line is not noisy. If there are many natural errors on a communications line, the impact of using the DES will be greater, i.e., the DES will multiply single bit errors, usually by a factor of 64. However, error detection protocols should minimize any effect of this phenomenon.

Question: To Keith Warble, Motorola

Is a preliminary specification sheet available for the Motorola Data Security Module?

Answer:

A copy of the preliminary specs can be obtained by writing to Mr. Durrell Hillis, Motorola, G.E.D., 8201 East McDowell Road, Scottsdale, Arizona 85252.

Question: To Clark Weissman, System Development Corporation

How do the DES and CRC (cyclic redundancy check) complement each other to prevent fraudulent modification of messages in a packet?

Answer:

The CRC error detection code or any similar polynomial code may be used in conjunction with the DES to provide message authentication. The error detection code should be generated on the plain message, then encrypted along with the message and the resulting cipher transmitted. The receiver should decipher the message, compute the error detection code from the received data and compare it with the error detection code transmitted with the data. This schema will ensure that a message cannot be modified even by an authorized person, without being detected by the receiver.

To prevent a "record and replay" threat from being used, a message sequence number must be generated, encrypted and transmitted with a message. The receiver must then verify that no messages have been lost, inserted or retransmitted.

Question:

Isn't there a problem with encrypting data for storage and not being able to read it later?

Answer:

If encryption is used to protect valuable data in storage, some method must be used to assure that it has been encrypted properly before it is stored, and then that it is also stored properly. Several alternatives are possible:

1. An independent device may be used to read the storage medium to assure that it has been written properly.

2. An independent device may be used to write a second copy which is then compared with the first.

3. In many data storage applications, the DES device may be completely duplicated and the results of the two independent devices compared before the data is written.

Question: To Robert Courtney, International Business Machines

Given a data storage environment, e.g., tape library or a shared disk system with combinations of both sensitive and non-sensitive data files, would you recommend protection of the sensitive data by

anonymity, i.e., not labeling it as being sensitive, or physically labeling the sensitive data and using rigid administrative procedures for protecting the sensitive data?

Answer:

Sensitivity is a "degree-to-which" sort of thing. It is rarely a simple binary variable. If you have relatively few sensitive tapes or disk packs which can reasonably be put in a vault, this is typically adequate given that the physical security is good. In general, sensitive data should be labeled as such and be given adequate protection. Security through anonymity generally is only adequate in a benign environment. In nearly all cases such as you describe, one should also examine the threats to the data from accidental or intentional modification and destruction. One usually finds that all stored data should be protected against these threats because the organization (company, agency) is often dependent on the availability of the data.

Question:

To what extent has NSA participated in the DES development?

Answer:

IBM developed the algorithm as published in the DES and submitted it to NBS during its public solicitations. NBS requested NSA to evaluate the algorithm for use in unclassified applications in the Federal Government. IBM designed the algorithm and NBS published it without any change.

Question:

What procedure will be followed and what criteria must be met in order for a waiver to be granted for a DES implementation in software?

Answer:

As stated in FIPS PUB 46, the DES is to be used by Federal agencies when encryption is desired and when the data to be protected is unclassified. The standard requires implementation of the DES in hardware for Federal usage. Software implementations in general purpose computers are not considered as complying with the standard. Federal agencies may waive the provisions of the DES after the conditions and justifications for the waiver have been coordinated with NBS. Software implementations for operational use must receive a waiver. However, software implementations for testing or evaluation do not require a waiver. The criteria to be considered when waiving the provisions of

the DES include the intended use of DES, how often it will be used, the impact on the system of a software implementation and the security required in the application.

Question:

The GSA recently responded to an agency's request to implement a secure telecommunications system to meet mandated confidentiality requirements by indicating that the Communications Act of 1934 outlawed interception and misuse of communications. GSA indicated that communications security for civilian agencies was therefore not needed. How should you interpret this in light of today's conference?

Answer:

The Communications Act of 1934 made the aural interception of communications illegal. A Federal Communications Commission investigation over the last several years has addressed the issue of interception of digital communications; the common interpretation is that the interception of digital communications does not violate the 1934 law. Incidentally, just making an act illegal does not necessarily stop it from occurring.

Question: To Clark Weissman, System Development Corporation

Doesn't the network security center approach to computer network security have a problem if an intruder gets the key used to protect future keys to be distributed within the network?

Answer:

If an intruder does obtain the device key used to distribute a working key to the device, it is obvious that the intruder can obtain all such working keys. Therefore, the device key must be given a very high level of protection and be changed on a regular basis, as well as whenever a security breach is suspected. The distribution and entry of device keys should be done by manual methods and the process must be protected.

Question:

In what time-frame is it expected that there will be sufficient demand for data encryption in commercial timesharing networks offering services to the Federal Government to warrant implementation?

Answer:

Data encryption will probably be requested as a feature in a time-sharing service for the Federal Government in two to five years.

Question: To Clark Weissman, System Development Corporation

What are the disadvantages of a network security center (NSC) for key generation and distribution?

Answer:

None has been built to date but the cost of an NSC will probably be high initially. There will be some overhead associated with the distribution of keys in the network. The security of an NSC must be very high. Maintenance of the data base of authorized users, terminals and computers, as well as their associated keys will be difficult and therefore costly. In addition, the normal costs of data base maintenance will be incurred at the NSC.

Question:

A recent paper by Professor Hellman of Stanford University has criticized the DES from various aspects. In particular, he claims that a characteristic of the DES can be used to cut the search time for an unknown key by 50% under a partially chosen plaintext attack. He also claims that the substitution tables are "fairly close to linear", that S-4 is 75% redundant and that the algorithm may contain a trap door. How were the substitution tables developed, are they truly random or do they have specific structure?

Answer:

The DES algorithm was reviewed by experts in encryption, including Professor Hellman, at a workshop held at NBS in September 1976. The characteristic of the algorithm identified by Professor Hellman is well known and can be used for various purposes. In particular, the characteristic is that if all of the inputs to the algorithm are complemented, the output is complemented. The chosen plaintext attack requires that a penetrator be able to collect not only matching plain and encrypted data, but also be able to collect matched plain and encrypted data that is the complement of the first data. This is not always possible. If an exhaustive search is made to find an unknown key, all of the possible keys must be potentially tested, but only half, namely 36 quadrillion, of the actual encryption operations must be performed. In actual work factor, the reduction is less than 50%. The characteristic is useful to implementors in that the encryption complementing devices may be easily tested during operation by simply complementing all of its inputs and being sure that the results of an encryption or decryption operation are also complemented.

The results of Professor Hellman's work show that the S boxes were not linear. No one at the September workshop could demonstrate the existence of a "trap door" in the DES algorithm. The designer of the

algorithm stated that the substitution tables are not random, that they indeed have structure based on a selected set of necessary and sufficient security criteria, and that a set was chosen to particularly minimize their implementation in LSI technology.

Question:

Technical questions on the following subjects were also posed: cost of DES device; mean time between failure; delay imposed on communication system; reduction in data transfer rate; increase in transmission errors; test method for devices?

Answer:

The answers to these questions will vary with many factors. The cost of an LSI DES device will depend on market volume, the technology used, the speed of the device and the yield of its products. Typical purchase prices may range from $50-$200 per LSI chip. When imbedded in a terminal, the price may range from 5-15% of the cost of the terminal. When implemented in a stand-alone encryption unit, the price will range from $1500-$4000.

Mean time between failure for most encryption units will be measured in years. Delay in communications will be measured in micro-seconds and reduction in data transfer rate will be negligible, and will often be used to detect accidental errors or intentional errors induced in the communication system.

The devices will be tested in various ways. Redundant DES devices may be used and output compared before encrypted data can be transmitted or stored. The complementary characteristic of the algorithm can be used to test an operational device. Loop-back tests can be used. Known test patterns can be used periodically. Independent devices can and should be used before critical data is stored in encrypted form for a long period of time.

NBS-114A (REV. 7-73)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET	1. PUBLICATION OR REPORT NO. NBS SP 500-27	2. Gov't Accession No.	3. Recipient's Accession No.

4. TITLE AND SUBTITLE

COMPUTER SCIENCE & TECHNOLOGY:

Computer Security and the Data Encryption Standard

5. Publication Date: February 1978

6. Performing Organization Code: 640.01

7. AUTHOR(S): Dennis K. Branstad, Editor

8. Performing Organ. Report No.

9. PERFORMING ORGANIZATION NAME AND ADDRESS

NATIONAL BUREAU OF STANDARDS
DEPARTMENT OF COMMERCE
WASHINGTON, D.C. 20234

10. Project/Task/Work Unit No.

11. Contract/Grant No.

12. Sponsoring Organization Name and Complete Address (Street, City, State, ZIP)

U.S. Civil Service Commission, 1900 E Street, NW.
Washington, D.C. 20415; and the
National Bureau of Standards, Washington, D. C. 20234

13. Type of Report & Period Covered: Conference Proceedings

14. Sponsoring Agency Code

15. SUPPLEMENTARY NOTES

Library of Congress Catalog Card Number: 78-1403

16. ABSTRACT (A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here.)

These proceedings include papers or summaries of presentations of the fifteen speakers who participated in the Conference on Computer Security and the Data Encryption Standard held at the National Bureau of Standards on February 15, 1977. Representatives from Federal agencies and private industry presented technical information and guidance with respect to computer security and the Data Encryption Standard. Subjects of the papers and presentations include physical security, risk assessment, software security, computer network security, applications and implementation of the Data Encryption Standard. The questions raised at the conference and their answers are included in the proceedings.

17. KEY WORDS (six to twelve entries; alphabetical order; capitalize only the first letter of the first key word unless a proper name; separated by semicolons)

Computer security; cryptography; Data Encryption Standard; encryption; key management; network security.

18. AVAILABILITY

[X] Unlimited

[ ] For Official Distribution. Do Not Release to NTIS

[X] Order From Sup. of Doc., U.S. Government Printing Office Washington, D.C. 20402, SD Stock No. SN003-003-

[ ] Order From National Technical Information Service (NTIS) Springfield, Virginia 22151

19. SECURITY CLASS (THIS REPORT): UNCLASSIFIED

21. NO. OF PAGES: 135

20. SECURITY CLASS (THIS PAGE): UNCLASSIFIED

22. Price: $3.00

USCOMM-DC 29042-P74